

ORIC OWNER



APRIL/MAY 1984 · ISSUE 7

TYPE IN EDUCATIONAL PROGRAMS
WE LIST THE SOFTWARE THAT'S AVAILABLE
PLAY PINBALL WITH YOUR ORIC

The very best in arcade action-

is brought to you by Tansoft – the specialists of the Oric-1 and Atmos Computers.

The games that people are fighting for are:-
Defence Force:

Pilot your space-craft through alien territory and fight off the enemy approaching from both sides. **48K. £7.95.**

Ultima Zone:

Shoot the Walkons, avoid the bouncing Brunes and battle your way through the satellite zone. **48K. £8.50.**

Rat Splat:

A revolting game.

Splat the rats in the sewer before they consume the cheese, and destroy the monsters before they consume you! **48K. £7.95.**



or let's get down to business!

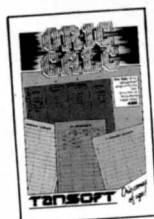
Tansoft produce a wealth of business and utility software to cater for every interest. The 'not so' games-minded people may be coming to grips with:-

Author: A powerful word-processor program which includes word-wrap and word-count. Ideal for letters and reports. **48K. £14.50.**

Oric Calc: A full spread-sheet program allows you to sum columns or rows and apply complex formulae. **48K. £14.50.**

Oric Cad: A computer aided design program for displaying, rotating, enlarging and shading objects on the screen. **48K. £9.99.**

For the full range of Tansoft software – which currently numbers over 20 titles – pick up a catalogue in your local Computer Store or contact Tansoft at Unit 1, The Techno Park, 645 Newmarket Road, Cambridge. Tel. Teversham (02205) 2261.



TANSOFT

Contents

APRIL/MAY 1984—ISSUE 7

COMMENT, Page 4

It's out with the old and in with the new

NEWS, Page 5

A successful show, Oric sponsorship and plenty more



BOOK REVIEWS, Page 6

The latest releases are assessed

CLUB SPOT, Page 7

There's a foreign flavour this time

PRACTICAL MACHINE CODE, Page 9

Part two of an easy to follow introduction to the complexities of Machine Code

"DEAR ORIC OWNER", Page 13

What you have to say to us, and what we say to that

DISASTER AREA, Page 14

Had problems with a recent listing. Here's what the listings should have had in them . . .

COMPETITION, Page 15

Win yourself a copy of The Hobbit, a classic adventure game



SOFTWARE FOR YOU, Page 16

Seven programs for you to type in

SOFTWARE LISTINGS, Page 19

The listings – and they're large enough to read without a magnifying glass

SIMPLE SENTENCES

PINBALL

CHARACTER DEFINER

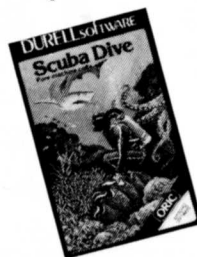
3D ROTATION

CHARACTER REDEFINER

SPELLBUZZ ANALYSIS

SOFTWARE REVIEWS, Page 39

Reviews of the software we've been enjoying for the last few weeks, and think you should be enjoying too!



SOFTWARE LISTINGS, Page 40

We bet you never thought there was as much software as this available for your Oric

QUICKIES, Page 43

Plenty of short programs for you to enter. Some are simply demonstrations, others are more useful

AND FORTHLY . . ., Page 46

Four new words defined

APPEND/RENUMBER, Page 47

A Machine Code program that gives far better file handling on the Oric computer. You can combine two programs, and renumber listings

HINTS AND TIPS, Page 52

Useful advice from other Oric owners to make your computing easier

NEXT ISSUE

We'll be interviewing the inventor of the Atmos, looking at the Microdrive, examining Oric's sound features and supplying all the best listings, reviews and latest news.

Cover illustration by Ainslie Macleod

Editor's Comment

Oric Owner

Editor: Kester Cranswick
Assistant Editor: Carolyn Groeneveld
Publisher: Paul B. Kaufman
Printers: Heffers Printers Ltd.
King's Hedges Road Cambridge



After two continuous years of producing magazines, the time has come for me to step down to allow someone who knows what they are doing to take over. That person is Kester Cranswick, who joins us from a well known photography magazine. His experience in this field, plus his long-standing Oric ownership, makes him the ideal person to develop and expand Oric Owner magazine.

As Managing Director of Tansoft, I now have the responsibility of ensuring the continuation of Oric Owner and the development of a wide range of software for the Oric. I shall submit pieces now and again, particularly on technical subjects such as Forth.

Kester and I will be available to meet you at all the major Computer Exhibitions this year and everyone is welcome to talk about all Oric related subjects. I shall look forward to meeting you.

Bye bye for now.

Paul B Kaufman

Publisher

When I bought my Oric, back in the early days, I never imagined it would

lead to me sitting in the editor's chair of Oric Owner. At that time I was the technical editor of a weekly photographic magazine, interested in learning about computers.

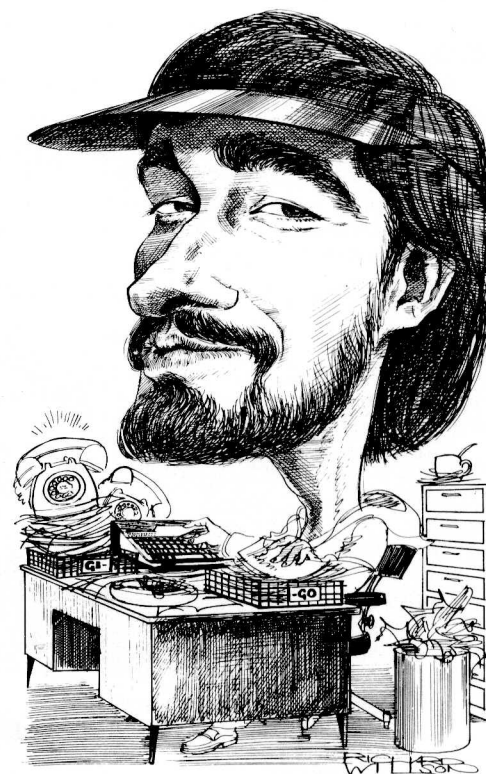
Since then I've learnt quite a bit about the Oric and computing in general. Still, I'm no expert, and make no claims to be. However, here at Oric Owner I'm surrounded by experts, and part of my job will be to put across information in a way that every Oric owner can understand.

I know that the information and programs that come in every day from you is of a very high calibre. I know because I've always read Oric Owner, and benefited tremendously from it.

In the coming months I hope to make Oric Owner even better than it has been. There are plenty of plans afoot, and many ideas to look into. So, if you have any ideas or suggestions for the magazine, let me know.

Kester

Editor



Oric Owner is published at bi-monthly intervals by Tansoft Ltd.
Units 1 & 2,
Cambridge Techno Park,
Newmarket Road
Cambridge
Tel: 02205 2261
Reg. No. 1632070
ISSN 0265-8488
Subscriptions: £7.50 for six issues (UK only). Overseas rates on request
Backnumbers: £1.20 (3 and 4 only)
© Tansoft Ltd.

RUNNING FOR ORICS



Oric are doing their bit to help Britain's athletes get to Los Angeles for this year's Olympic games. Oric put up prizes for a national Run for Britain, held on the 18th March at Wisley Park, Surrey.

The prizes went to the runners who raised the most money from their sponsors. Two 16K Oric-1 computers are to be awarded to the top money raisers, one to an under 16 runner, one to an over 16 runner. And, just in

case some runners felt they'd not got enough sponsorship, all runners returning their sponsorship money by 21st April will be eligible for a special prize draw. First out of the barrel in that wins a brand new 48K Oric Atmos. With the lure of a home computer like the Oric, the national organizers hope to go a long way towards achieving the total target of £1.5 million pounds needed by the British Olympic hopefuls.

TANSOFT DEVELOPS LINK-WORD LANGUAGE LEARNING SYSTEM

LANGUAGE MASTERCLASS, the very first computerised Link-Word foreign language learning system, was launched recently, at the LET '84 Show by British software and publishing house, Tansoft Limited, of Cambridge.

Claimed to provide the ability to communicate at a basic level in a selected language within ten to twelve hours – around a 70% reduction over conventional book learning methods – MASTERCLASS is essentially aimed at British businessmen who travel abroad, couriers, tourists and home-study students. It is only compatible with the new British 48K Oric Atmos microcomputer and its forerunner, the Oric-1 48K, and is now available in French, German, Italian and Spanish versions.

LET '84 LAUNCHES

What's in store for '84 was the theme of the inaugural Leisure Electronics Trader show at the Heathrow Penta hotel in mid-February. The three day, trade only show, attracted 120 exhibitors, enough to fill three floors.

Oric took space to show off the new Atmos, the restyled printer and the yet to be released Micro Disc. There was plenty of Oric software to be seen too.

Tansoft had a second version of Chess, this time with synthesised voice commands. They also demonstrated their latest games and the Author word processing package.

Arcadia were showing a sample of Pasta Blasta, the first part of a Pasta Wars trilogy. Ocean had rigged up an old Space Invaders arcade game to demonstrate Hunchback. Durell were showing Harrier Attack on a five foot screen and Salamander were enthusing

about a new racing game, Classic Racing.

With the launch of the Atmos, there seems to be a growing commitment to Oric software on the part of major software houses. There are an awful lot of Oric owners out there just waiting for the right programs to come along. 1984 is looking set to be a great year for Oric software.



TERRY SHURWOOD JOINS ORIC BOARD FROM MATTEL



Terry Shurwood (43) has been appointed to the board of Oric Products International Limited, the Ascot based

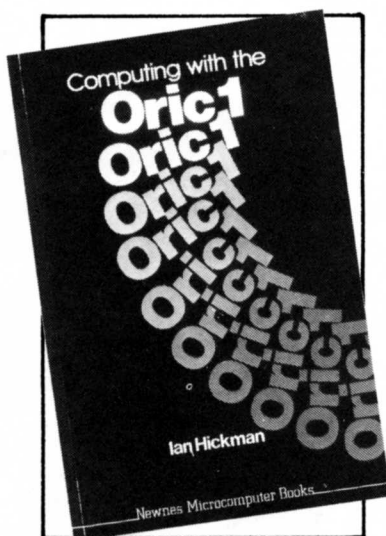
electronics company who last month launched the Atmos 48K micro computer.

Mr Shurwood joins Oric as Sales and Marketing Director, taking over from Peter Harding who assumes responsibility for new vertical markets including Viewdata.

A former National Account Manager for ITT, General Sales Manager for Ingersol Electronics and, most recently, Sales and Marketing Director for Mattel Electronics (UK) Ltd, one of Mr Shurwood's responsibilities will be the restructuring of Oric's sales force and improving support to the company's 1000 UK micro computer dealers.

COMPUTING WITH THE ORIC-1

Ian Hickman
Newnes Technical Books
£6.95



The Oric manual has come in for its fair share of (justified) criticism. This book is not so much a rewritten manual as a complementary volume. It sets out to fill the gaps of the original manual.

It is written without reference to the Atmos, so takes the bugs of the original Oric into account. The MCP40 printer is dealt with too.

So as not to alienate novices, the author starts with a look at the history of computers, explains common terms and how to get the Oric running. Sections on BASIC programming, colour, graphics, editing and mathematics follow. They are well written but if you've persevered with the manual, there's little new to be gleaned from them.

A chapter on Boolean logic is certainly valuable as that's a real gap in the manual. Then it's back to strings, more graphics and sound commands, with a few helpful tips here and there. Good demonstration programs are used throughout to illustrate certain features and points.

So far, so so. Now come the good bits. There's a chapter on tape handling, one on improving BASIC program structures and a lengthy section on Machine Code that's as good an introduction as we've seen anywhere. A chapter on printers is also worth reading too, as is the appendix on the Centronics interface.

For the beginner, this is a fairly good book, though the final chapters may well be confusing. For the more advanced user, the first chapters are unnecessary and the later sections, very handy. The book thus falls between two interest groups. Depending on your experience, it may be what you want, or it may not. Just like the curate's egg, it is good in parts.

METEORIC PROGRAMMING

John Vander Reyden
Melbourne House, £7.95



There's no shortage of programs to enter into your Oric. This collection of 30 listings is one of the most balanced around.

The usual game variants are included – Frogger, Bomber, Lunar Lander, Invaders and so on. They are all quite straightforward with few frills. Other games include a short

adventure, a three dimensional maze and some rather more original games.

More useful programs include basic payroll and sales analysis listing, a synthesizer listing, a working clock and a drawing aid program. And, for Machine Code programmers, the last section of the book has some useful explanations and utilities. These are a Machine Code monitor, a line renumber and a block line delete.

Each program is illustrated with a picture of the screen display. That gives an idea of whether the program is worth entering in. There are also good explanations of program structures, useful subroutines and variables. The budding programmer will certainly learn plenty from this book.

The programs contain a few bugs, chiefly the dreaded ELSE bug. A thoughtful feature is the indication of the lengths of empty strings – an idea we wish other program lists would follow.

If you are on the lookout for a book of programs that will teach as well as entertain, this is certainly worth getting. We rate it highly.

SIXTY PROGRAMS FOR THE ORIC-1

Erskine/Walwyn/Stanley/Bews Pan, £4.95

Produced in collaboration with Personal Computer News, this is a good value book of program listings. With 60 programs, it will keep you busy for weeks.

The programs are listed with nothing more than a brief description of the game. There are no tips, no screen displays and the only way to establish the value of a program is to laboriously type it in.

All the usual programs are

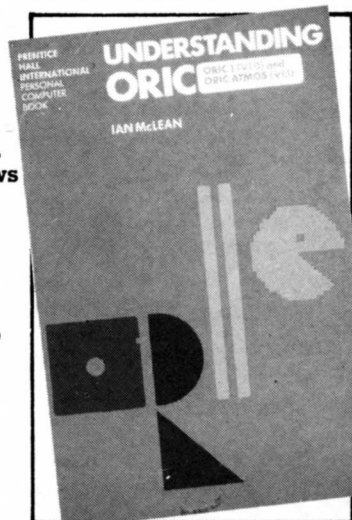
listed, along with a few novel ones. The most ambitious is a three part, 20 page game called Evolution. Other interesting listings are Options – to help you decide between a number of options, Holiday Expenses, a simple payroll calculator, a biorhythms listing and a moving message displayer.

The listings are rather lengthy, enabling the programs to be of a high standard. Bugs are uncommon – the publishers will supply corrections to Alien, Hangman and Minelay listings. Other errors will be your fault!

Whatever your game interest, there will be a few programs here for you. Take the trouble to type them in and you'll be royally entertained. At just £4.95, this book offers excellent value for money, if you want a good collection of games that you could improve on. Other books may educate you – this one is purely for amusement. It does that exceedingly well.

UNDERSTANDING ORIC

Ian McLean
Prentice/Hall £7.95



Were we to recommend just one book to the Oric owner, this would be the one. For a

user wanting information on their Oric, at whatever level, Ian McLean's book is a must.

For a start, it has full details of the Atmos, how it differs from the Oric-1 and how the new commands operate. Should you be thinking of upgrading to an Atmos, reading this book will detail what you'll benefit from.

The book starts at the very beginning – unpacking the computer. Experienced users will be able to gloss over these sections, though they are well written and very well illustrated, even if the style is a little condescending.

Short programs are given throughout, encouraging the reader to use his computer. Screen pictures of what should be displayed also help the reader too.

BASIC is tackled in a helpful manner too, with lengthy explanations of commands, examples and programming ideas. Word

and number handling is covered effectively. The sound section is extremely useful in its thoroughness and makes sense of the manual's instructions. The graphics chapters are equally good.

A chapter of miscellanea deals with attributes, POP, PULL, GRAB and RELEASE commands plus the file handling features of the Atmos. Then there's a chapter on printers (Oric and others), followed by a short introduction to Machine Code.

The back of the book contains numerous appendices – ASCII codes, BASIC commands, Assembly Language instructions and so on.

All in all, this book should be bought by every Oric owner, and adopted as a manual. It is a little short on Machine Code, but for the BASIC programmer, there's not a better book around.

CALLING ALL ADVERTISERS



The Oric Owner is the only magazine devoted to the Oric-1 and Atmos computers.

Virtually every person who buys an Oric will get this magazine. If you are selling Oric related products can you afford to miss this chance to reach the market you are aiming for?

For full details of our advertising rates contact Carolyn on Teversham 2261 and see if she can help you.

How many other Oric owners do you know? If you know a few, why not get a club going? We'll provide free publicity to readers, through the magazine, and help with any queries you may have.

For instance, if there are any Oric owners in the Harrow area, James Shields would be delighted to hear from you. He wrote to us with details of his Oric outfit and wanting to contact other owners. If you'd like to talk Oric with James, his address is 182 Welbeck Road, West Harrow, Middlesex HA2 0RX.

Moving further afield, a number of clubs have been established on the continent. Here are the contact addresses . . .

Belgium

Oric Club Verviers
Mr Philippe Mawet,
52 Rue Waoury,
4502 Queue-du-bois,
Belgium

Belgium

Mr Kaisin,
Rue de l'Idylle 19,
1080 Bruxelles,
Belgium

Belgium

Oric Computerclub Floppy
p/a Mathieu Doumen,
Dennenlaan 7,
3460 Maasmechelen

Holland

Nedoric,
Dhr. G. Tax,
Gasthuisring 24,
5041 DS Tilburg,
Nederland

If you're travelling abroad, why not contact these clubs?

If any Oric owners wish to set up a club, give us some details and we'll happily pass on information for you.

U.K.

Cardiff

12 Tregarth Court,
Creigiau,
Cardiff

Kent

c/o Roger Pyatt,
23 Arundel Drive,
Orpington,
Kent
Tel: 66 20281



Classifieds

This is the place to sell your excess software and accessories. Oric Owner classifieds will reach thousands of Oric owners who may be interested.

Classified ads cost just 15p per word, up to a maximum of 30 words. Just write your ad clearly and legibly on a postcard and send it to Oric Owner, Classifieds, Unit 1 & 2, Techno Park, Newmarket Road, Cambridge.

Oric-1 48K TAPE COPIER

makes backup copies of any standard Oric-1 Software package. (NB users must not infringe copyright).

Copies basic, machine code including protected autorun. Easy to use, repeat copies, instructions. Send £5.50 for cassette and instructions to:

A. Johnson, 11 Bedburn Avenue, Wear View Est, Sunderland, Tyne and Wear.

For the ORIC-1 and now also for the ATMOS

Educational software . . .

Story Book - A challenging game that develops reading skills. Suitable for any level from early reader onwards. You choose the text, the game is to reconstruct it by guessing the words. £4.95

Picture Book - A colourful spelling game for young children. Build up pictures by typing words and seeing the objects appear "by magic". £4.95

Personal finance . . .

Account Book - The money management program. Budgeting and record keeping by automatic double entry. Produces full reports on screen or printer. Easy to use and menu driven throughout. £12.50

- Softbacks -

Send to:

Softbacks, Dept. TS, PO Box 257, Watford, WD1 3LQ.
Please state model of computer. (Account Book is for the 48K ORIC-1 only.)

Send sae for further details.

DEDUCT 50p from total for more than one program.

DUCKWORTH HOME COMPUTING

All books written by Peter Gerrard, former editor of *Commodore Computing International*, author of two top-selling adventure games for the Commodore 64, or by Kevin Bergin. Both are regular contributors to *Personal Computer News*, *Which Micro?* and *Software Review* and *Popular Computing Weekly*.

EXPLORING ADVENTURES ON THE ORIC 48K by Peter Gerrard

This is a complete look at the fabulous world of Adventure Games for the Oric Computer. Starting with an introduction to adventures, and their early history, it takes you gently through the basic programming necessary on the Oric before you can start writing your own games.

Inputting information, room mapping, movement, vocabulary - everything required to write an adventure game is explored in detail. There follow a number of adventure scenarios, just to get you started, and finally three complete listings written specially for the Oric, which will send you off into wonderful worlds where almost anything can happen.

The three games listed in this book are available on one cassette. £6.95

Other titles in the series include *Sprites & Sound on the 64*, *12 Simple Electronic Projects for the VIC*, *Will You Still Love Me When I'm 64*, *Advanced Basic & Machine Code Programming on the VIC*, *Advanced Basic & Machine Code Programming on the 64*, as well as *Pocket Handbooks for the VIC, 64, Dragon, Spectrum and BBC Model B*.

Write in for a descriptive leaflet (with details of cassettes).



DUCKWORTH

The Old Piano Factory, 43 Gloucester Crescent, London NW1 7DY
Tel: 01-485 3484



SPECTRUM 16-48K

Scuba Dive (48K only)
Harrier Attack
Jungle Trouble

COMMODORE 64

Scuba Dive
Harrier Attack

ORIC 16-48K

Scuba Dive
Harrier Attack
Galaxy 5
Starfighter (48K only)

ATARI

Harrier Attack

DURELL software

available from

**W.H. SMITH · LASKYS · GREENS AT
DEBENHAMS · BOOTS · COMPUTERS FOR
ALL · MARTINS · SPECTRUM CENTRES
and many other
retail outlets**

**or order direct from
CASTLE LODGE
CASTLE GREEN
TAUNTON
SOM. TA1 4AB**

**TRADE ENQUIRIES TO:
SOFTWEAR
COMMUNICATIONS LTD
MARTECH HOUSE
4, BAY TERRACE
PEVENSEY BAY
EAST SUSSEX
TEL: (0323) 768456**

MAIL ORDER

SPECTRUM software £5.50 all other software £6.95
Price includes postage and packing

TYPE OF COMPUTER:	
CASSETTE TITLES:	
1)	
2)	
3)	
NAME:	
ADDRESS:	

Overseas customers please allow £1.00 extra

PRACTICAL MACHINE CODE

PART 2

Machine Code is not as complicated as it looks. D. Sinfield continues his explanations of it from last issue. This time, he looks at screen scrolling.

The program to move the top line of the screen in the first part of this article (Oric Owner 6) showed that machine code programming isn't just for whizz kids who think in code. Admittedly it is a long way removed from BASIC and it does require some preparatory work before entering a program. A proper monitor or assembler/disassembler can relieve the programmer of a lot of the drudgery, especially in working out offsets and looking up opcodes on a chart, but is the extra effort worth it? To answer this question let's now compare the speed of machine code with that of an analogous program in BASIC.

The incrementing and decrementing of the Y index done to point to the character position along the screen line can be replaced by a FOR/NEXT loop. This also takes care of the conditional branch we used to check that the whole line was done. The LDA/STA pairs used in the code can be replaced with PEEKs and POKEs. The resulting BASIC looks like this:

```
10 B = PEEK (48079)
20 FOR A = 48078 TO 48042 STEP-1
30 N = PEEK(A)
40 POKE A + 1, N
50 NEXT
55 POKE 48042, B
Add 5 FOR C = 1 TO 500
60 NEXT
to run program 500 times
```

I have converted all the hex addresses to decimal to make the program run as quickly as possible. Enter and run both programs to compare the speed. You should find there's a massive difference. Even using a clumsy BASIC/FOR/NEXT loop to CALL our routine 500 times it only takes 2.7 seconds. The BASIC takes in the region of nearly four minutes to do the same thing. The amount of memory used can also be critical. The first two lines of the BASIC program use more memory than the whole machine code routine.

I hope you're now convinced that it is worth looking a little more into machine code programming. First, define what you want to do because if you don't know where you're going any road will take you there. The line scrolling program can be extended to scroll the whole

screen. The initial flowchart is in Fig. 1. Notice that the flowchart doesn't get any more complicated because there is not need to detail the line scroll. We've already done it.

Skip down to 'have we done whole screen?'. For the line scroll program to work we want to enter the address of the position of the left hand side of the line in 0000 and 0001. We can use this to decide when the bottom of the screen has been reached. The last line's address is BFBA so the program should end when it arrives at the decision with 0000 containing BA and 0001 containing BF. Have a look at the extended program in Fig. 2. It should be familiar to line 15 but note that the RTS in line 16 of the line scrolling program has gone. That was the return from Subroutine to get us back into BASIC (RTS is not always used like this). Following the familiar bit are six lines (16 to 21) that decide whether the bottom line has just been moved to the right.

These are new and deserve a flowchart of their own (Fig. 3). It looks a bit complex but if you follow it through step by step you'll see that the only way to reach the end is to have a 'yes' to both decisions. If the answer to either question is 'no' there are one or more lines left to do. Now, look at the assembly code in Fig. 2 to see how this has been achieved. Line 16 is another Load Accumulator instruction in a new mode. Line 1 is an example of LDA immediate, line 6 uses the same instruction but indirect indexed and now in line 16 we have zero page LDA. All it means is that the accumulator should have the value of zero page address 00 (the low byte of the line pointer) transferred to it in preparation for making the decision. In line 17 the instruction is new but the mode, indicated by the hash mark (#) should ring some bells. The instruction is CoMPare. It's very useful because it actually does the subtraction sum required without affecting the contents of the accumulator. How can doing a sum but not producing an answer be helpful? The solution lies in the way the 6502 decides whether or not to branch when it comes to a conditional instruction.

The 6502 has a special byte in its architecture called the Processor Status Word, often referred to as the PSW. This contains eight single byte 'flags'. It is these bits that are tested by the Branch instructions. The flags are set and cleared as the result of various instructions. The details of the flags that will be affected by a particular instruction are given in appendix K of the Oric manual. The flag we are interested in for all the conditional branches in this program is the zero flag. It is set if the result of an instruction is zero and cleared if the result is non-zero. BNE causes a branch if the zero flag is cleared and BEQ branches if the zero flag is set. Taking an example from the program so far: line 13 causes the program to branch back to line 8 if the zero flag was cleared by the result of line 12 being other than zero. This may seem a convoluted way to go about it but later when the programs get more complicated we will find that transactions not only in the indices X and Y and the

Machine Code

accumulator but anywhere in RAM can result in changes in the flags and thereby be used as the criteria for branches.

So the compare instruction does a subtraction and although it does not produce an answer it alters the relevant flags. We test the low byte of the address pointer against BA. If the result is not zero then the bottom of the page has not been reached, there's no need to check the high byte and the program skips to the updating of the pointer. If, however, the low byte is equal to BA then it is necessary to check the high order byte. This is done by lines 19 and 20 and if the result of the comparison of the high byte of the pointer (at 0001) is zero then we want to exit from the program. The Branch at line 21 points to the RTS that returns control to BASIC.

Another of the flags is the carry flag and the next section of the program, dealing with updating the screen pointer, uses the carry. The start of the next line on the screen has an address 28 (hex) higher than the line above. To move the pointer down we want to add 28 to the address pointer. This sounds straight forward but remember that we can only deal with one byte at a time and addresses are two bytes wide. The first three lines are no problem – they start at BBAA, BBD2 and BBFA and all we need to do is add 28 to the least significant byte. But, how do we calculate that the next line starts at BC22? Both bytes of the address have changed.

Let's deal with the easy one first. Look at lines 23 to 25 in the program. The carry flag is in an unknown state – it may have been set back in BASIC before this routine was entered. Clear Carry makes sure that the carry flag is zero before we start the calculations. The next instruction is one we haven't seen before: Add with Carry. The hash mark indicates that the instruction is in immediate mode so the number following is the value to be used in the transaction. So, 28 is to be added to the accumulator. If you look at the program you'll find that at this point in the program the accumulator holds the value that was at 0000 – the low byte of the pointer. For the first iteration of the routine the accumulator holds the value AA. Line 24 does the following operation:

Accumulator (AA) + Operand (28) + Carry flag (0)

It puts the result in the accumulator. Do a bit of Hex arithmetic and you'll find that the answer to the above calculation is D2 which sure enough is the low byte of the pointer for the next line. Line 25 should be familiar – it stores the accumulator in the zero page address (0000) we are using for the low byte of the pointer. On the third pass through the program the accumulator will hold FA and the above sum will be like this:

Accumulator (FA) + Operand (28) + Carry Flag (0)

The result of this calculation is 122. The accumulator is only one byte wide so clearly this result will not fit. This is where the carry comes in. It, in this case, extends the

accumulator by one bit. I'm not going into the intricacy of binary, so believe that the leading 1 in 122 is represented by the most significant bit of this binary number:

```

      1   0010   0010
      └───┬───┬───
      1   2   2
  
```

The eight least significant bits (the eight on the right) can go into the accumulator. The one that doesn't fit goes into the carry. The carry need only be one bit wide because the largest result by adding two bytes and the carry bit is 1FF (FF + FF + 1 = 1FF).

Now, go back to the program in hand. On the third pass through, the accumulator and carry flag will look like this after each instruction:

	accumulator	carry	0000	0001
Line 22	FA	not known	FA	BB
Line 23	FA	0	FA	BB
Line 24	22	1	FA	BB
Line 25	22	1	22	BB

The carry has to be added to the high order byte of the pointer held at 0001. Line 26 loads this byte into the accumulator and line 27 adds just the carry, because the operand is zero. The result is stored at the address of the high order pointer (0001).

The process looks like this:

	accumulator	carry	0000	0001
line 26	BB	1	22	BB
line 27	BC	1	22	BB
line 28	BC	1	22	BC

So, after line 28, the pointers are as we want them and we know (almost) all there is to know about double precision arithmetic.

All the we now require of the program is to go back to the point at which we start to do the line. There is an instruction in the 6502 set called JuMP which does the job. The addressing mode of JuMP most used is absolute. This means that following the one byte of instruction two bytes of address are given as the destination of the jump. One thing to remember about any of the absolute mode instructions is that the low byte of the address must be given first. When it comes to coding the instruction 'jump to 9808' it looks like this:

assembler	machine code
JMP 9808	4C- opcode, JuMP
08	operand (destination address)
98	

Machine Code

Fig. 1.

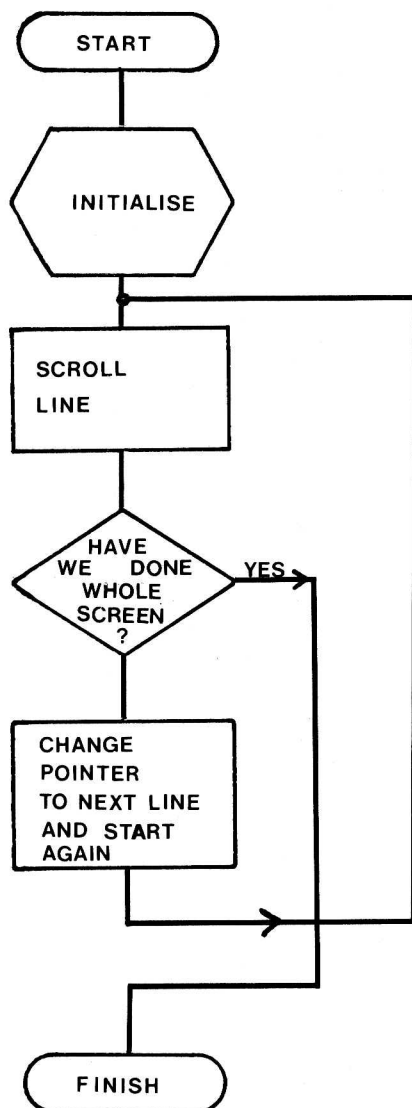
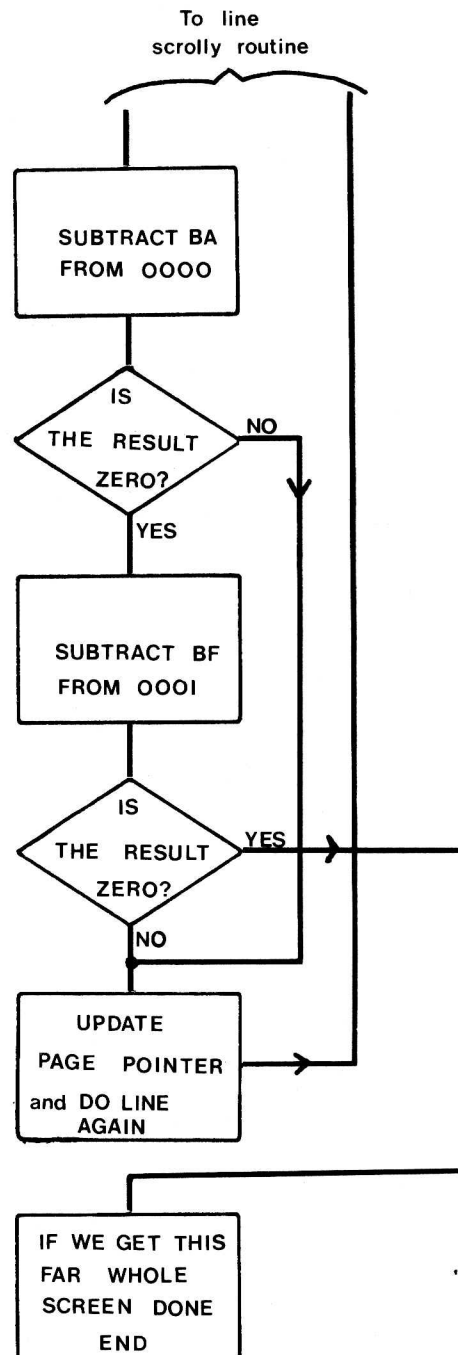


Fig. 3.



Machine Code

Another thing to remember about absolute jumps like this is that the code is no longer relocatable i.e. it cannot be moved about in memory without altering the JuMP instruction. If this is going to be a disadvantage (if you're using the FORTH assembler for instance) and you don't know what the absolute address of the destination is going to be, then a conditional branch can be used as long as you make sure the condition is going to be true. An example using a flag that we are familiar with would be:

```
CLC
BCC relative offset
```

To put this into words: CLear Carry then Branch if Carry Clear. It is obvious that the branch will always be carried out.

The final RTS is only ever executed from the branch at line 21 and it returns control to BASIC.

That's the whole program but by no means the end of the story. As it stands the program can be used to scroll right from any line downwards but it will always run to the bottom of the page. You may remember how to do this from the earlier line scrolling program. The top of screen pointers can be set up from BASIC and the program entered at 9808. How can flexibility be improved? One approach might be to have the bottom of

page address in memory somewhere and instead of doing the comparing in lines 17 and 20 in the immediate mode, use zero page mode. This means only four bytes of the program need changing:

Line no.	old assembly	new assembly	Machine Code
17	CMP #BA	CMP 02	C5 02
20	CMP #BF	CMP 03	C5 03

If you do use this then it is essential that the following rules are observed:

1) 0002 must contain the low order byte of the address of the left hand side of the screen of the line you want to be the last of those moved. 0003 must contain the high order byte.

2) The line pointed to by 0002 and 0003 must be lower down the screen than the line pointed to by 00 and 01.

If either of these conditions are not fulfilled then the program will mess about with memory that's not part of the screen and will probably crash. This brings to light one thing to watch in Machine Code – if your finger slips in BASIC and a word is spelled wrong, BASIC traps the error. No such safeguards operate in Machine Code – you're on your own talking directly with the CPU!

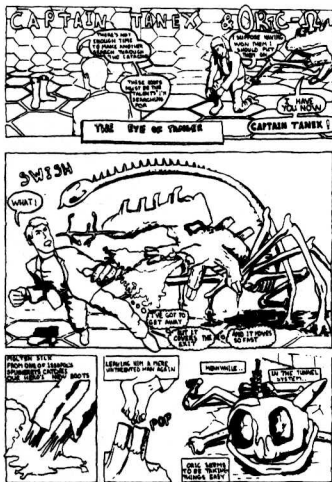
L N.	A C	A	M C	R	Fig. 2.
1	LDA #AA	9800			
2	STA 00				
3	LDA #BB				
4	STA 01				
5	LDY #25				
6	LDA (00), Y				
7	PHA				
8	DEY				
9	LDA (00), Y				
10	INY				
11	STA (00), Y				
12	DEY				
13	BNE F7				
14	PLA				
15	STA (00), Y	9817			
16	LDA 00	19	A5 00		
17	CMP #BA	1B	C9 BA		
18	BNE 06	1D	D0 06		
19	LDA 01	1F	A5 01		
20	CMP #BF	21	C9 BF		
21	BEQ 10	23	F0 10		
22	LDA 00	25	A5 00		
23	CLC	27	18		
24	ADC #28	28	69 28		
25	STA 00	2A	85 00		
26	LDA 01	2C	A5 01		
27	ADC #00	2E	69 00		
28	STA 01	30	85 01		
29	JMP 9808	32	4C 0898		
30	RTS	35	60		

This should be familiar
as the line scrolling
routine

Get low byte of pointer
Compare it to last line address
If it's not equal then skip high byte comparison
Get high byte of pointer
Compare it to bottom line
If it's equal skip to Exit
Get low byte
Make sure carry = zero
Update pointer
Put it back
Get high byte of pointer
Add the carry from the low byte
Put it back
Do next line
Go back to BASIC

"Dear Oric Owner"

Cartoons



I hope you won't mind a few friendly criticisms of Oric Owner. Per page it is costlier than more general magazines, so readers may be fussier.

My two greatest hobbies are comics and computers, in that order. Please, please stop Captain Tanex, who is doing great damage to both, in that order.

Please stop those interviews telling us how great the Oric is and how clever you all are. Please stop printing those 'gosh', 'wow' letters too. I have already bought my Oric, and you don't have to keep selling it to me.

Could we have an article explaining the PLAY command in detail? I must confess that having read and reread the relevant chapter in the manual several times, I'm still rather uncertain about how it functions.

How about a full list of system variables, or two – one for each BASIC release? It's silly that your contributors tell you how the machine functions.

Martin Hagström, Denmark

We take your points in the right spirit. How do the rest of you feel about interviews and cartoons? As for the article suggestions, watch these pages. Ed

Interfacing

I have now my Oric-1 (48K) for about 1 year, and I'm very pleased with it.

Lots of my friends have bought a Commodore 64 and they aren't so happy with it (problems with the recorder, HI Resolution etc. . .). Still, I wish to make one complaint about Oric (overseas). The Oric Owner magazine is only available in Antwerp. I have found my first issue 2 weeks ago (OCT/NOV).

Now to the point of my letter: I'm not so interested in computer games but I like programs like Word-1, Assembler, Disassembler, Databanks, Forth etc. . .

Also I wish to use my micro for external use, therefore I'm asking you for a copy of a design which a I/O interface and how to use it. I already made my own joystick interface and it works very well.

On the DIN plug (back of the Oric) for the recorder there is a relay-contact. How can it be used?

Can you also send me a list of disc-drives (floppy) that can be directly connected to the Oric-1 without interface. BECU Phillippe, BELGIUM.

If anybody out there has a design for an I/O interface, let us know. We'll happily publish details. As for a disc drive unit, the Oric drives are promised for the near future.

The relay contacts of the DIN socket are for remote operation of a cassette recorder, if it has the remote facility. You'll have to use a seven pin DIN cord. Ed

Composite Video

There are problems with the 'Composite Video From The Oric 1 circuit featured in Oric Owner 5.

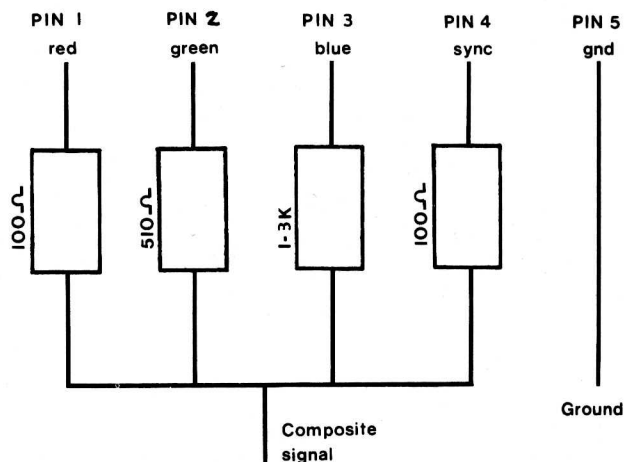
Firstly, it will not work unless the sync pulse is added.

Secondly, by putting equal resistances in the R, G and B lines, red green and blue will all come out in the same shade of grey. Using software written for a colour monitor, this could be a problem! The

screen would be a uniform grey.

The following circuit diagram does the job quite well. It is worth experimenting with different resistances before doing the final assembly.

D. J. Wyper, Glasgow



Notice of Renewal

Before you throw away the envelope this issue of Oric Owner came in, look at the address label. If there is an R next to your name, it means your subscription should be renewed, and this is the last issue you'll get on your current subscription.

Fortunately, a new subscription won't cost as much as the old. We've been able to cut costs, so the next six issues will cost just £7.50, instead of £10.00. European subscriptions are £10.98 for six issues. Contact us for overseas rates. You'll find a form at the back of this issue. Alternatively, if you don't want to cut up the issue, drop us a line with a cheque made out to Tansoft Ltd. We look forward to having you with us for the next six issues.

ORIC OWNER SUBSCRIPTION FORM

Please send me the next 6 issues

I enclose a cheque for (cheques payable to Tansoft Ltd)

Name.....

Address.....

..... Postcode.....

Back issues are available for £1.20 each. If you missed your first free issue contact your dealer or Oric Products International who will supply you with one.

TANSOFT Tansoft Ltd, Units 1 & 2
Cambridge Techno Park Newmarket Road, Cambridge CB5 8PB

Disaster Area

AND FORTHLY...

Forth enthusiasts may be struggling with the Money Manager listing given in Oric Owner of December/January.

Two screens were omitted from the listing. Here they are...

Screen 1

```
0 (PRNTR PACKAGE FOR
  MONEY MANAGER-
  CONTROLS &
  HEADINGS)
1 : TAB 9 EMIT :: PRINIT
  PON 27 EMIT 64 EMIT ::
  BIGCH 14 EMIT :
2 : U/L 27 EMIT 45 EMIT 1
  EMIT :: NU/L 27 EMIT 45
  EMIT 0 EMIT :
3 : SETTABS 27 EMIT 68
  EMIT 11 EMIT 28 EMIT 38
  EMIT 48 EMIT
4 58 EMIT 68 EMIT 0
  EMIT :: SMCH 20 EMIT :
5 : LICAP TAB BIGCH U/L
  " LIST OF ITEMS" NU/L
  SMCH :
6 : LIHEAD U/L " TITLE"
  TAB TAB " CR" TAB "
  DR" TAB
7 " START" TAB " FINISH"
  TAB " PERIOD" NU/L :
8 : FRCAP TAB BIGCH U/L
  " FORECAST REPORT"
  NU/L SMCH :
9 : FRHEAD1 " FROM "
  NOW .DATE " TO " FIN
  .DATE " WITH A"
10 " STARTING BALANCE
  OF #" BAL D@ .STERL :
11 : FRHEAD2 U/L " DATE"
  TAB " TITLE" TAB " CR"
  TAB
12 " DR" TAB "
  BALANCE" NU/L :
13 : CR/DR AMOUNT + D@
  3 PICK C/D + C@ 68 = IF
  TAB " #' .STERL
14 ELSE " #' .STERL TAB
  THEN :
15 —
```

SCREEN 2

```
0 (PRNTRPACK/MONEY MGR-USER WORDS LIST-ITEMS
  & FORECAST-REPORT)
1 : .REC DUB 16 TYPE TAB DUP CR/DR TAB DUP START +
  .DATE TAB
2 DUP FINISH + .DATE TAB .PERIOD :
3 : PRXSAC DUP NEXT + .DATE TAB DUP 16 TYPE TAB
  DUP CR/DR TAB
4 DUP AMOUNT + D@ ROT C/D + C@ = IF DMINUS
  THEN BAL D@ D+
5 OVER OVER BAL 2+ ! BAL ! DUP ROT ROT " #" .STERL
  0< IF
6 " DR" ELSE " CR" THEN :
7 : LIST-ITEMS RECORD DROP PRINIT SETTABS CR LICAP
  CR CR LIHEAD
8 CR TOP MAXRECS 0 DO RECORD DUP C@ 32 = 0= IF
  CR .REC ELSE DROP
9 THEN DOWN LOOP CR " END LIST" CR POFF :
10 : FORECAST-REPORT CLS RECORD DROP FROMTO?
  PRINIT SETTABS FRCAP
11 CR CR FRHEAD1 CR CR FRHEAD2 SET1ST BEGIN
  EARLIEST DUP NEXT +
12 FIN LATER WHILE DUP DUP NEXT + SWAP FINISH +
  LATER IF DUP CR
13 PRXSAC THEN UPDNXT REPEAT DROP CR "END
  REPORT" CR POFF :
14 FORTH DEFINITIONS LATEST 12 +ORIGIN ! HERE 28
  +ORIGIN !
15 HERE 30 +ORIGIN ! HERE FENCE ! " ***" .S
```

ORIC-1 FIG-FORTH

14 Oric Owner

CINDERELLA

Cinderella, the adventure in Oric Owner 5 has a number of bugs in the listing. Here are the corrected lines.

```
400 IF C$="RUN"THEN
  GOSUB 2570
440 IF (B$="N" OR
  LEFT$(ZS,@)="GO N")
  AND S%(P%,1)<>0
  THEN P% = S%(P%,1) :
  K=1
1690 REM INVENTORY
  ROUTINE
1200 DATA on the main road
  leading North/South,
  32,35,0,0
1111 DATA in a hallway. The
  staircase leads up from
  here., 0,0,3,1
2220 .. haven't
```

ORIC COPY

If you use the Oric Copy program from issue 5 you will find that the MCP40 printer prints a _ instead of a £ sign. Mr P. R. Mitchell of Ayrshire has a modification that will result in a # being printed. At #040D, replace 207BF5 with 4C6D04.

Then insert the following

```
#046D C95F [ CMP #5F
#047F F006 [ BEQ #0477
#0471 207BF5 [ JSR #F57B
#0474 4C1004 [ JMP #0410
#0477 A923 [LDA #23
#0479 4C7104 [JMP #0471
```

WORD-1

Word-1 was a simple word processor program published in Oric Owner October/November. It had some pretty major bugs. Reader Mr R. Entwistle of Sheffield has managed to get it running almost as described, though the sub-editing routine still needs a little work. Here are his suggested corrections.

command. PLOT commands cannot plot in column 0 when in the TEXT mode.

```
460 GOTO 120
¼ in 1030 and elsewhere
should be the ↑ symbol.
⅛ should be a < sign and
⅜ a > sign.
```

```
1115 ..... GOTO 1122
```

Add LORES 0 after every CLS

```
8050 IF P=1 THEN 8200
```

TELEPHONE DIRECTORY

Problems arose with the Telephone Directory listing of the Feb/March Oric Owner. You will not be able to store the name-number arrays created with a simple CSAVE command.

If you have an Oric-1 this program will not run without extensive modification to give

an array saving and array loading routine. However, if you own an Atmos, you could use the STORE routine. Replace CSAVE in line 530 with STORE. You will also need to build in a routine to use the RECALL command of the Atmos to reload the array into memory.

ATMOS CONVERSION

Atmos owners wanting to use listings for the Oric-1 computer won't find conversion very difficult. With a BASIC program, all you need to do is to add one to

the first, X co-ordinate of any PLOT command. Remember to alter variables that relate to this figure too. Programs will then work as they're meant to work.

THE COMPETITION- 50 HOBBITS TO BE WON

Why spend £14.95 on The Hobbit when you might win one for the cost of a stamp? We have 50 copies of the game to give away. Solve the puzzle and one of them could be yours.

Entry is simple. The 12 names that are from or related to The Hobbit. They could be written forwards, backwards, horizontally or vertically. All you have to do is find the names, write them on the back of a postcard or sealed envelope and send them to us.

The address is

Hobbit Competition
Oric Owner
Unit 1 & 2,
Techno Park,
Newmarket Road,
Cambridge

Entries must arrive by last post, 30th April, 1984. All correct entries will be put in a box and the 50 winners chosen. Only one entry per person is allowed, and you must remember to put your address on the entry.

The Game

Playing The Hobbit is an experience. You are plunged into the fantasy world of J. R. R. Tolkien's Middle Earth, peopled by trolls, dwarfs, wizards and wargs. Your task is to recover the Dragon's treasure. You'll meet many of the characters from the book, converse with them, have them as allies or maybe fall out with them. As adventure games go, The Hobbit is a masterpiece. It is only fitting that it took the inaugural Golden Joystick award for Best Strategy Game of the Year.

The game is supplied with a copy of the original book. To solve the adventure, use the book as a guide. The game follows the book's plot

L	A	G	A	O	H	L	P	T	W	A	S	N	Q
S	O	U	S	N	E	I	K	L	O	T	P	X	M
U	E	R	B	M	I	E	Z	Q	D	A	C	N	I
F	L	A	D	N	A	G	R	O	J	L	Y	I	P
T	Y	T	R	O	L	L	S	R	L	S	B	R	A
D	U	L	W	V	F	K	H	M	U	L	L	O	G
S	E	O	F	P	S	T	Y	R	T	L	A	H	C
R	L	M	I	T	E	R	H	O	B	B	I	T	A
J	R	Q	P	D	V	V	M	E	U	I	O	D	Z
B	O	S	D	Y	R	H	J	K	R	V	B	M	K
S	N	I	G	G	A	N	O	B	L	I	B	S	W
E	O	H	P	Q	W	C	F	T	Q	I	N	G	H
G	K	L	S	R	D	S	N	I	L	B	O	G	T
E	S	U	A	N	A	I	R	D	A	F	I	W	S



pretty closely. The display uses both text and graphics. graphics illustrate key locations, while the text tells you what's going on. There are no sound effects to annoy neighbours.

While writing The Hobbit, the team devised some new programming techniques that put The Hobbit in a class of its own. The first is that the program accepts inputs very close to ordinary English. The language is called English. It knows over 500 words and

few hard and fast rules need be followed to communicate with the computer. No adventure game has ever been this user friendly.

Another breakthrough is Animation. The characters you'll meet act independently as the game proceeds. They make their own decisions and influence the course of the game. You'll have to learn how to influence other characters. You can talk to them quite freely in English.

Other useful features of The



Fact sheet

Company: Melbourne House

Established: 1977

Offices: UK, USA, Australia
Mail Order address: 131
Trafalgar Road, Greenwich,
London SE10

Customer Service: Church
Yard, Tring, Herts HP23 5LU
Sales & Marketing: Cable
Yard House, Cable Yard,
Richmond, Surrey TW10 6TF

Oric software: The Hobbit
£14.95

Oric books: Meteoric
Programming, £5.95

Other books: Guide To
Playing The Hobbit, £3.95

Computer And Video Games
Book Of Adventure, £5.95

Hobbit are a printer toggle so you can get a hard copy record of how a game proceeds, a pause feature, and the facility to save a game to tape for later use. The instruction booklet explains commands thoroughly, though playing the game is a continual learning experience.

There are many solutions to The Hobbit, and many, many traps. For those who get really stuck, Melbourne House will supply tip sheets from their Customer Services department. There is also a book, 'The Guide to Playing The Hobbit,' by David Elkam and priced at £3.95. It details one possible solution. There's no need to go to the lengths of one player who range London four times from New Zealand to get out of a tricky situation.

If you are already a Hobbit player, and have some tips or advice for other players, send it to Oric Owner. If there are enough tips, we will start a Hobbit Help feature.

PINBALL

Once upon a time every good pub, cafe and entertainment arcade had at least one pinball machine. Now the space invader and its successors have replaced the pinball. If you regret the passing of that particular aspect of your youth, this program from M. Russell of Keighley, West Yorkshire is well worth entering.

This pinball game is called Demon's Den. Up to four players can compete for top score, with three balls each. Keys 1 to 9 control how hard the ball is struck at the beginning, and how hard the flippers strike. The higher the number, the harder the ball is struck. There's no tilt though, but go easy on your keyboard.

Just like the real pinball machines, the Oric uses sound and colour to good effect. There are bonuses too, and it will take a good deal of practice to get a high score. We've managed to get 124,268 and are sure somebody out there can beat that.

The program itself is pretty straightforward, and in BASIC. It is well structured and a good example of how logical thinking can make even the vagaries of the pinball machine be realised by the Oric.



CHARACTER REDEFINER



The ability to redefine characters is one of the major tools of a good games programmer. Here's a short program that lets you redefine any character pixel by pixel. You'll be able to build exactly the character you want.

Instructions in the program explain how it works. You are presented with a coloured matrix of eight by six cells. Using the cursor control keys and a simple on/off feature, you can turn any pixel in the matrix on or off to construct the character you want.

You are also prompted to enter the character you want redefined, or its ASCII number, and given a repeat option at the end. Characters will remain redefined until your Oric is reset.

This program has no means to store data figures that could be used in other programs to duplicate the character you have redefined. However, from the shape on the matrix, it shouldn't be too difficult to work out the code for each of the eight lines of the character. The six columns, from the left, have the values 32, 16, 8, 4, 2 and 1. Add together the column values for all pixels that are turned on in a line and you have the data number for that line. Use the eight such numbers from each matrix, and the code for the character, in future programs.

Now, can anyone redefine their Oric to print Japanese characters?

SIMPLE SENTENCES

We all know that computers make excellent educational tools. They are infinitely patient, won't make mistakes and can be a great asset in the home. Here's an educational program S. W. Lucas of Cheadle Hulme, Cheshire, designed to teach young children simple word recognition.

What it does is to print out a short sentence with a word missing. Above the printed sentence, four boxes are drawn. They each contain a possible solution. An arrow moves below the boxes and the child presses a key when the arrow is below what is thought to be the correct solution.

That's the signal for a loud right or wrong sounds, with the correct answer printed out. After ten sentences, a score is given, a rating and prompts for entry of the child's name. A repeat option is also given.

REMs throughout the listing explain how the program is constructed. There are up to sixty sentences to choose from, and more could be added. You can also change the number of questions per round.

The program is aimed at five to seven year olds. So, type it in, sit them down at the keyboard and let Oric play teacher for a while.



ANALYSIS

Computers aren't only for fun and games. There is a serious side to them too. This program puts the Oric to work analysing sales figures. It is by Rob Sibthorpe of Fordingbridge, Hants.

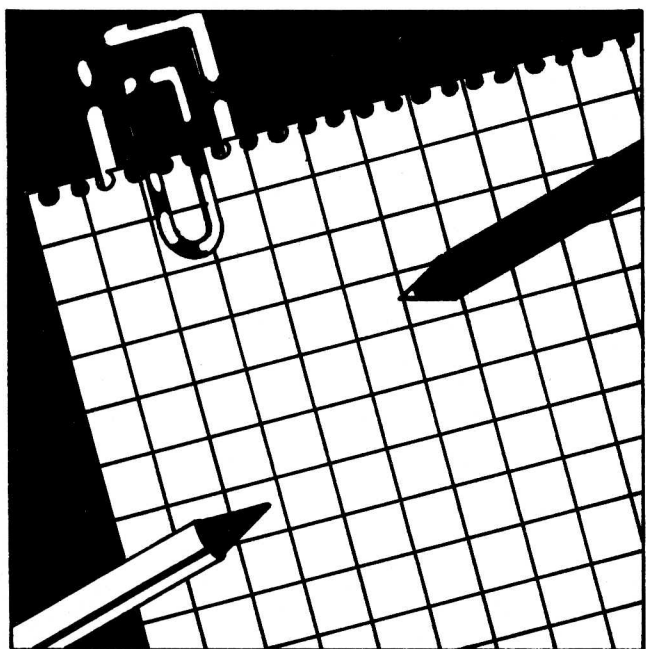
When RUN, you are asked for a title of the analysis, how many entries there are to be and column headings. There can be up to ten columns for each entry, but the last must be cost.

You then make the entries, following the on screen prompts. To reserve space for later data, enter 0.

A menu appears next on screen. These are the options...

1. EXTRACT COSTS
2. LIST PARTICULAR ENTRY
3. LIST COLUMN VARIATIONS
4. LIST LIKE ENTRIES
5. LIST ALL ENTRIES
6. LIST SPARE ITEMS
7. CHANGE ENTRY
8. STOP

Follow the prompts from here and the program runs like a well oiled machine. It isn't possible to save data to tape, but this program should enable an astute analysis of costs, sales or whatever. As a BASIC business program, it is very useful and does indeed demonstrate that the Oric belongs in the office as well as the home.



SPELLBUZZ

Education seems to be a major theme in this issue. Our second program of an educational nature comes from Rob Sibthorpe of Fordingbridge in Hampshire.

Written for seven to ten year olds, Spellbuzz is a spelling test. It runs from a dictionary of 100 words. These could be added to or changed to suit different age groups.

When RUN, three alternative spellings of a word, in a random order, are listed. The child is prompted to enter the correct spelling. The correct spelling is then printed out double height and a smile or frown face indicates a right or wrong answer. After ten words a rating is given accompanied by sound effects.

An unusual feature of this program is a permanent, flashing title display above the text area. Lines 250 to 295 do that. Otherwise the program is pretty straightforward BASIC. Let the kids have a go with it and their spelling should get much better!



3D ROTATION

We've all seen computers doing three dimensional graphic representations of objects. Here's a program for your Oric that shows a three dimensional cube rotating in space on the HIRES screen. It is adapted from a program by D. F. Round that appeared in the Tansoft Gazette two years ago. The updater is Mr M. J. Bunce of Rugby, Warwickshire.



Your inputs determine the size of the cube and how it rotates. Be careful with the inputs though, as the wrong type of figures may crash the program.

You'll first be asked for the eye position. Stick to the 150 to 200 range. Smaller values may give too large a cube, and a DRAW command will give an error message.

Next input are the three direction cosines, separated by commas. Suggested inputs are

1,0,0 – pure rotation about X axis
0,1,0 – pure rotation about Y axis
1,1,0 – rotation about both axes

Finally you'll be asked for the total degrees to be rotated and the step, the values separated by a comma. For a pure rotation about one axis, a 90° total is all you need to see the whole effect. Remember that 360° gives a full rotation.

For the step, we'd suggest five degrees. Anything less will mean very small changes between consecutive drawings. Larger steps will mean less drawings to reach the total rotation.

Experiment with different figures and you'll see cubes in ways you never thought possible. Now, who can do a rotating pyramid?

Software Listings

INDEX

SIMPLE SENTENCES

A program that teaches word recognition, for young children, Page 20

PINBALL

If you didn't misspend your youth, here's your chance to become a pinball wizard, Page 24

CHARACTER REDEFINER

Redefine characters quickly, easily, so you can make your own games characters or symbols, Page 28

3D ROTATION

A graphics display program that shows solid objects rotating in space, Page 29

SPELLBUZZ

Improve spelling abilities with this useful educational listing, Page 31

ANALYSIS

Business people will appreciate this sales analysis program. It does everything but save data, Page 34

SIMPLE SENTENCES

```

1 REM ***SIMPLE SENTENCES ***
2 REM ***SUITABLE FOR 5-7 YEAR OLDS**
3 REM *** WRITTEN BY S. W. LUCAS ***
4 REM *** <C> August 1983 ***
5 AA$="Simple sentences"
6 AB$="written for the ORIC1"
7 AC$="by S.W. Lucas "
10 HIRES
12 PAPER3:INK4
16 REM ROUTINE TO PRINT THE TITLES ON T
HE HIGH RESOLUTION SCREEN
20 CURSET20,20,3
30 FOR A = 1 TO LEN(AA$)
32 CHAR ASC(MID$(AA$,A,1)),0,1
33 CURMOV10,0,0
35 NEXT A
36 CURSET 20,50,3
38 FOR A = 1 TO LEN(AB$)
39 CHAR ASC(MID$(AB$,A,1)),0,1
40 CURMOV10,0,0
41 NEXT A
43 CURSET 20,80,3
44 FOR A = 1 TO LEN(AC$)
45 CHAR ASC(MID$(AC$,A,1)),0,1
46 CURMOV 10,0,0
47 NEXT A
100 REM THE ARRAY A$(X,Y) HOLDS THE SEN
TENCE TOGETHER WITH THE FOUR
110 REM ALTERNATIVE ANSWERS. THE CORREC
T ANSWER IS HELD AS A NUMBER IN
115 REM B$(X) WHICH IS READ FROM THE ST
H DATA ITEM ON EACH DATA LINE
120 DIM A$(60,5),B(60),C(10)
130 GOSUB 270
140 GOSUB1280
150 GOSUB 1710
180 GOSUB1330
190 SC=0
195 IF (B$="N" OR B$="n") THEN END
200 RUN
210 END
211 REM THE REMAINDER OF THE PROGRAM CO
NSISTS OF SUBROUTINES ONLY
220 REM THE FIRST SUBROUTINE IS TO
230 REM READ THE DATA INTO THE 2-DIMENS
IONAL ARRAY A$(X,Y)

```

```

240 REM EACH QUESTION HAS 4 ALTERNATIVE
S
250 REM FOLLOWED BY THE NUMBER OF THE C
ORRECT ALTERNATIVE
260 REM THIS ANSWER IS HELD IN THE ARR
Y B(X)
270 FOR X =1 TO 60
280 FOR Y = 1 TO 5
290 READ A$(X,Y)
300 NEXT Y
310 READ B(X)
320 NEXT X
330 DATAJill found a ---- in the sand.
,shell,shade,shop,ship,1
340 DATAYou can write with a ---,pin,pa
n,pen,tin,3
350 DATADad said he had --- the garden,
dog,log,dug,bag,3
360 DATABen said "I hit the ball with a
---",bag,bat,bin,bun,2
370 DATAThe baby sleeps in a ---,cat,cu
t,nut,cot,4
380 DATAYou fish with a ---,red,rod,rid
,rug,2
390 DATATom --- the ball into the pond.
,hit,hut,hot,1
400 DATAMum will eat the ---,ban,bin,bu
n,ben,3
410 DATADad put the --- in the bin,ten,
tan,tip,tin,4
420 DATATom will --- in the bus.,set,si
t,sun,sum,2
430 DATAMum put on her --- to sit in th
e sun,hut,hat,hit,his,2
440 DATABen had --- in the park,fun,sun
,fin,fan,1
450 DATAJill took her --- to the shop.,
big,bag,bug,beg,2
460 DATATom ---- for the bus,runs,pens,
puts,dive,1
470 DATAMum will ---- a cake for me,bak
e,rake,lake,fake,1
480 DATAShe met Tom in the ----.,lark,b
ark,park,mark,3
490 DATAJill likes to ---- her bike.,ri
de,bide,side,tide,1

```


500 DATA Bill likes to --- the animals i
 n the zoo., see, saw, sun, sip, 1
 510 DATA They want to go to the ----, lar
 k, mark, dark, park, 4
 520 DATA You can ride in a ----, far, bar,
 car, tar, 3
 530 DATA The ----- like to eat grass., sh
 eet, shell, ships, sheep, 4
 540 DATA Bill sat on a -----, chair, chas
 e, chin, chips, 1
 550 DATA Dad likes to play with the ---,
 fat, mat, cat, pat, 3
 560 DATA It goes ---- at night., bark, dar
 k, mark, park, 2
 570 DATA Tom likes to read a ----, look, t
 ook, book, cook, 3
 580 DATA I would like to --- a cake, pat,
 tap, eat, bat, 3
 590 DATA Mum poured the milk out of a --
 -, jug, lug, tug, bug, 1
 600 DATA The boy will go to --- tonight.
 , bed, bud, bug, bun, 1
 610 DATA Paul wants a --- for a pet, log,
 dog, fog, jog, 2
 620 DATA The boy played with a ----, kite
 , mite, site, bite, 1
 630 DATA We put the plant in a plant ---
 ., lot, tot, dot, pot, 4
 640 DATA John played cricket with a ----
 , bull, bell, ball, bill, 3
 650 DATA The drink of tea was very ---, h
 ut, hit, hat, hot, 4
 660 DATA The number after nine is ---, t
 in, tan, ten, one, 3
 670 DATA The boy wrote with a ---, pin, p
 an, pun, pen, 4
 680 DATA That man is very ----, tall, til
 l, tell, toll, 1
 690 DATA The dog chases a ---, cut, cat, c
 ot, sit, 2
 700 DATA I like to play the game of ----
 and seek., hide, tide, ride, side, 1
 710 DATA Jim swam to the ---- of the bat
 h., ride, side, hide, tide, 2
 720 DATA We went to the ---- to buy a ca
 ke., ship, shop, shut, shed, 2
 730 DATA John went for a ride on a ----
 ., trail, tramp, train, trap, 3
 740 DATA The number after two is ----, ,
 throw, thank, think, three, 4

750 DATA I don't want to ---- down the s
 tairs., tall, hall, fall, call, 3
 760 DATA Mary had lots of --- playing wi
 th her ball, sun, fun, bun, run, 2
 770 DATA Dad was --- when he had to go h
 ome., sad, fad, dad, lad, 1
 780 DATA Dad had to ---- his broken bike
 ., send, fend, mend, lend, 3
 790 DATA We had to ---- at the side of t
 he road., stand, sting, start, stray, 1
 800 DATA Paul went to the park to ----, ,
 play, plug, plus, plum, 1
 810 DATA Paul was ---- for school yester
 day., gate, hate, rate, late, 4
 820 DATA Mum had to ----- her money., spi
 ll, spend, spray, spell, 2
 830 DATA Dad cut his ---- this morning.,
 chain, chair, chin, chap, 3
 840 DATA Mary put the food in a ----, fi
 sh, dish, wish, wash, 2
 850 DATA I couldn't ---- up this morning
 ., cake, rake, wake, take, 3
 860 DATA I drank a ----- of water., class
 , glass, grass, brass, 2
 870 DATA Dad undid the ---- with his key
 ., rock, dock, sock, lock, 4
 880 DATA We have a fish ---- in our hous
 e., rank, bank, tank, sank, 3
 890 DATA Jill looked at her ---- in a mi
 rror., race, face, lace, pace, 2
 900 DATA We planted a rose in the ----, ,
 soil, foil, boil, toil, 1
 905 DATA The dog likes to ----, mark, bar
 k, park, lark, 2
 910 DATA I missed the ---- bus and had t
 o walk home., last, mast, cast, vast, 1
 920 DATA She sees Tom in the ----, toad,
 load, road, sad, 3
 930 RETURN
 950 REM THE NEXT SUBROUTINE DRAWS 4 BOX
 ES ON THE HIRES SCREEN
 960 REM TO HOLD THE 4 ALTERNATIVE ANSWE
 RS
 965 REM THE BOXES WILL HOLD WORDS OF UP
 TO 5 LETTERS ONLY !
 970 HIRES
 971 REM RESELECT THE COLOURS OF THE HIR
 ES AREA
 972 PAPER7

```

973 INK4
974 YY=20
975 FOR Q= 1 TO 4
977 CURSET YY,50,1
980 DRAW 39,0,1
990 DRAW 0,30,1
1000 DRAW -39,0,1
1010 DRAW 0,-30,1
1020 YY=YY+52
1030 NEXT Q
1040 RETURN
1220 REM THE FOLLOWING SECTION OF THE P
PROGRAM
1230 REM CHOOSES A RANDOM SELECTION OF
1240 REM TEN QUESTIONS OUT OF THE 60 PO
SSIBLE QUESTIONS
1250 REM THE ARRAY C(X) IS USED TO HOLD
THE
1260 REM NUMBER OF THE QUESTIONS TO BE
ASKED
1270 REM TO CHOOSE MORE QUESTIONS ALTER
THE NEXT LINE
1280 FOR X=1 TO 10
1290 LET C(X)=INT(RND(1)*60)+1
1291 REM THE NEXT LINE ENSURES THAT TWO
IDENTICAL QUESTIONS
1292 REM DO NOT COME NEXT TO EACH OTHER
1293 IF C(X)>1 AND (C(X)=C(X-1)) THEN G
OTO 1290
1300 NEXT X
1310 RETURN
1330 SC=0
1340 REM FIRST SET THE SCORE =0
1350 REM THEN SET THE 10 QUESTIONS WHIC
H WERE CHOSEN AT RANDOM
1357 REM FROM THE 60 ALTERNATIVES
1360 FOR TT = 1 TO 10
1380 REM HIRES IS USED LIKE CLS TO CLEA
R THE HIGH RES SCREEN
1390 HIRES:
1399 REM THEN DRAW THE 4 BOXES AGAIN
1400 GOSUB 970
1410 GOSUB 1470
1420 GOSUB 1760
1430 GOSUB 2140
1440 NEXT TT
1450 GOSUB 2400
1460 RETURN
1470 G=B(A)
1480 A=C(TT)

```

```

1490 REM A =C(TT) HOLDS THE NUMBER
1493 REM PRINT THE QUESTION AT THE BOT
OM OF THE SCREEN IN THE TEXT AREA
1494 REM CHOOSE A COLOR FOR QUESTION
1495 INK4:PRINT A$(A,1);
1500 REM OF THE CURRENT QUESTION
1510 REM G=B(A) HOLDS THE ANSWER
1520 REM TO THE CURRENT QUESTION
1570 REM THE NEXT SECTION PRINTS THE
1580 REM WORDS IN THE BOXES
1582 PP=22:REM SET X POSITION OF CURSOR
FOR THE FIRST WORD
1584 REM THE NEXT SECTION PRINTS THE 4
ALTERNATIVE WORDS
1585 FOR P= 2 TO 5
1590 CURSET PP,60,3
1595 N$=A$(A,P)
1600 FOR Z= 1 TO LEN(N$)
1610 CHAR ASC(MID$(N$,Z,1)),0,1
1620 CURMOV Z,0,0
1630 NEXT Z
1635 PP=PP+52:REM MOVE CURSOR FOR NEXT
WORD
1640 NEXT P
1650 RETURN
1670 REM THIS SUBROUTINE IS TO DEFINE
1680 REM THE CHARACTERS TO BE USED
1690 REM AS A POINTER
1700 REM TO POINT TO THE WORDS
1710 GF$=CHR$(129)
1720 GG$= CHR$(94)
1730 GH$= CHR$(124)
1740 RETURN
1760 D=32
1770 E=100
1780 REM PROCEEDURE TO MOVE ARROW ACROS
S
1790 REM THE SCREEN . TO POINT AT THE 4
BOXES IN TURN
1800 REM UNTIL ANY KEY IS PRESSED
1820 REPEAT
1830 REM D = X POSITION AND E = Y POSIT
ION
1840 REM OF THE ARROW
1850 ZAP
1860 REM SOUNDS PING EACH TIME THE ARRO
W MOVES
1900 CURSET D,E,3
1910 CHAR94,0,1
1920 CURMOV 0,8,3

```

```

1930 CHAR124,0,1
1950 WAIT 150
1951 REM CHOOSE APPROPRIATE WAIT TIME F
OR AGE OF PUPIL. SMALL VALUE = HARD
1960 REM THEN ERASE THE POINTER
1970 CURSET D,E,3
1980 CHAR 94,0,2
1985 CURMOV0,8,3
1988 CHAR124,0,2
2000 D=D+52
2010 REM MOVE ARROW
2015 REM F= POSITION ARROW STOPS IN
2020 REM IF ARROW GOES OFF SCREEN, RESE
T IT TO POSITION 1
2040 IF D>198 THEN D=32
2050 REM RESET CHOICE WHEN ARROW GOES O
FF SCREEN
2070 S$=KEY$
2075 REM TESTS FOR ANY KEY BEING PRESSE
D
2080 UNTIL S$<>" "
2085 REM SET VALUE OF F
2090 IF D=32 THEN F=4
2095 IF D=84 THEN F=1
2100 IF D=136 THEN F=2
2110 IF D=188 THEN F=3
2120 IF F>4 THEN F=4
2125 IF F <1 THEN F=?
2130 RETURN
2140 HIRES
2150 PAPER7
2160 INK 4
2180 REM CHECK FOR CORRECT RESPONSE
2190 IF F=B(A) THEN GOSUB2210 ELSE GOSU
B2310
2200 RETURN
2209 REM SUBROUTINE FOR THE CORRECT RES
PONSE
2210 SC=SC+1
2220 REM IF RIGHT ;- INCREMENT THE SCOR
E
2230 REM NOW PRINT ON THE SCREEN RIGHT
2235 RR$="right"
2236 CURSET 89,75,3
2240 FOR Z= 1 TO LEN(RR$)
2242 CHARASC(MID$(RR$,Z,1)),0,1
2245 CURMOV8,0,0
2248 NEXT Z
2250 REM NOW GIVE A SOUND FOR RIGHT ANS
WER

```

```

2251 FORX=255 TO 1 STEP -1
2255 SOUND1,X,15
2260 NEXT X
2262 FORX=1 TO 255
2264 SOUND 1,X,15
2265 NEXT X
2267 SOUND *1,0,0
2300 RETURN
2305 REM SUBROUTINE FOR WRONG ANSWER
2310 WW$="wrong"
2315 REM TO WRITE WRONG ON THE SCREEN
2320 CURSET 89,75,3
2325 FOR Z= 1 TO LEN(WW$)
2330 CHARASC(MID$(WW$,Z,1)),0,1
2335 CURMOV 8,0,0
2336 NEXT Z
2340 WW$="It was "+A$(A,B(A)+1)
2341 REM TO PRINT THE CORRECT ANSWER
2344 CURSET 50,150,3
2346 FOR Z= 1 TO LEN(WW$)
2348 CHARASC(MID$(WW$,Z,1)),0,1
2349 CURMOV 8,0,0
2350 NEXT Z
2353 REM SOUND EFFECT FOR WRONG ANSWER
2354 FORY=1TO3:FOR X= 150 TO 250
2355 SOUND1,X,15
2356 NEXTX,Y
2357 SOUND1,0,0
2360 WAIT 200
2370 REM PAUSE TO LET CHILD READ WHAT T
HE ANSWER SHOULD HAVE BEEN
2390 RETURN
2400 TEXT
2410 REM ROUTINE FOR PRINTING RESULTS
2420 LET A$=""
2430 IF SC>5 THEN A$="good"
2440 IF SC>7 THEN A$=" very good"
2450 PAPER 7
2460 INK 0
2470 PRINT:PRINT:PRINT:PRINT" you
r score was ";SC;" out of 10"
2480 PRINT:PRINT:PRINT"What is your nam
e ";
2490 INPUT NA$
2500 PRINT:PRINT:PRINTCHR$(129);"Thank
you for playing ";CHR$(132);NA$
2510 PRINT:PRINT:PRINTCHR$(130);"Do you
want to play again?"
2520 GETB$
2530 RETURN

```


PINBALL

```

5 GRAB
10 REM >>>>>>>>>><<<<<<<<<<
20 REM > Pinball Demons Den <
30 REM > M Russell 170683 <
40 REM > for Oric Owner <
45 REM >>>>>>>>>><<<<<<<<<
50 POKE618,10
60 H(2)=1
100 GOSUB40000
105 CLS
110 GOSUB50000
120 DOKE616,514:PRINT:PRINT" HOW":PRINT
" MANY":PRINT"PLAYERS":PRINT"1 TO 4":GET
Q$
130 DOKE616,514:PRINT:PRINT" ":PRIN
T" " :PRINT" " :PRINT" "
140 IFQ$<"1"ORQ$>"4"THENSOUND1,200,15:S
OUND1,0,15:GOTO120
142 FORN=1TOVAL(Q$):B(N)=3:S(N)=0:NEXT
150 FORN=1TO4:A$="SCORE "+RIGHT$(STR$(N
),1):PLOT31,N*3+N-2,A$:NEXT
160 FORN=0TO4:PLOT32,N*4+4,"000000":NEX
T:PLOT32,18,"BONUS!"
170 PLOT32,24,"BALL 1"
175 B=1
180 FORP=1TOVAL(Q$)
181 GOSUB50000
185 POKE616,(P-1)*4+2:PRINT:POKE617,31:
PRINTCHR$(140)
190 X=28:Y=23:J=0:K=-1
200 PLOTX,Y,41
210 PLOT2,0,"PRESS":PLOT1,1,"ANY KEY":P
LOT2,2,"1 TO 9":PLOT1,3," FOR "
220 PLOT1,4,"STRENGTH":PLOT1,5,"OF FIRS
T":PLOT1,6," SHOT " :GETS$
230 FORN=0TO6:PLOT1,N," " :NEXT
240 IFS$<"1"ORS$>"9"THENZAP:WAIT50:SOUN
D1,0,15:GOTO210
250 S=VAL(S$)+20
260 PLOT1,1,"FLIPPER":PLOT2,2,"KEYS":PL
OT2,3,"1 TO 9":PLOT1,4,"TO VARY"

```

```

270 PLOT1,5,"STRENGTH":PLOT1,6,"OF SHOT
"
280 K$=KEY$:IFK$="1"ANDK$<="9"ANDD=0TH
ENGOSUB1000
290 IFD>0THENGOSUB2000ELSEPLOT15,23,"
"
300 O=SCRN(X+J,Y+K)
310 IFO=32THEN500
320 IFO=64THEN4000
330 IFO=46THEN4030
340 IFO>34AND0<41THEN4060
350 IFO=63THEN4090
360 IFO=45THEN4170
370 IFO=42THEN190
380 IFO=60THENGOSUB620:GOTO500
390 IFO=62THEN4230
400 IFO=79THEN4320
410 IFO=59THEN4400
420 IFO=61THEN4400
430 IFO=47THEN4470
440 IFO=58THEN4500
450 IFO=93THEN4540
460 IFO=91THEN4600
470 IFO=123THEN4650
480 IFO=125THEN4700
500 PLOTX,Y,32
510 X=X+J:Y=Y+K:PLOTX,Y,41
520 S=S-A:IFS=0THENGOSUB3000
527 IFY=24THEN532
530 GOTO280
532 PLOTX,Y,32
533 K$=KEY$
535 FORU=5TO70:FORZ=1TO100STEP1:SOUND1,
Z,10:NEXTZ,U:WAIT10:SOUND1,0,0
536 PLOT30,(P-1)*4+2,32
537 B0=0:PLOT32,20,"000000"
538 PLOT31,22," " :S(P)=S(P)*H(2)
:GOSUB7000:H(1)=0
539 H(2)=A
540 NEXTP
550 B=B+A

```

```

560 IFB=4THEN700
570 PLOT37,24,STR$(B)
580 GOTO180
600 J=A:RETURN
610 K=A:RETURN
620 J=-A:RETURN
630 K=-A:RETURN
640 K=0:RETURN
650 J=0:RETURN
700 PLOT14,12," GAME OVER"
702 FORN=1TO4:IFS(N)>HC0)THENH(0)=S(N)
703 NEXT
705 U=280
707 PLOT16,15,"Press Y":PLOT16,16,"to p
lay":PLOT17,17,"again"
710 K$=KEY$:U=U+1:IFU=300THENGOSUB800
715 IFK$=""THEN710
720 IFK$="Y"THEN105
730 IFK$="N"THENEND
740 ZAP:GOTO710
800 PLOT31,1," HIGH":PLOT31,2,"SCORE "
810 M=S(1):P=1:S(P)=H(0):GOSUB7000
820 WAIT70
830 PLOT31,1,"      ":PLOT31,2,"SCORE 1"
835 WAIT155
840 S(P)=M:P=1:GOSUB7000:U=0:GOTO710
1000 D=A:F=A
1010 T=VAL(K$)
2000 ONDGOTO2010,2020,2030
2010 PLOT15,23,"(      #":D=D+F:RETURN
2020 PLOT15,23,"' (      #$":D=D+F:RETURN
2030 PLOT15,23,"&' (      #$%":F=-1:D=D+F:RET
URN
3000 J=0:K=1:S=S+1
3010 RETURN
4000 IFK=-AANDJ=ATHENK=A:GOTO500
4005 IFK=-ATHENGOSUB620:GOTO500
4010 IFJ=ATHENGOSUB610:GOTO500
4020 GOSUB620:GOTO500
4030 IFJ=-ATHENGOSUB610:GOTO500
4040 IFK=-ATHENGOSUB600:GOTO500
4050 GOSUB600:GOSUB610:GOTO500
4060 PLOTX,Y,32:X=X+J:Y=Y+K:PLOTX,Y,41
4070 K=-A:PLOTX,Y,0
4071 IFO<38THENJ=INT(RND(1)*2)-A

```

```

4072 IFO>37THENJ=INT(RND(1)*2)
4075 S=S+(T*3)
4080 GOTO500
4090 IFJ*K<>0THEN4130
4100 J=0-J:K=0-K:GOTO500
4130 IFK=-AANDSCRN(X,Y-A)=32THENGOSUB65
0:GOTO500
4135 IFJ=AANDK=-AANDY=3THENK=A:GOTO280
4136 IFJ=-AANDK=-AANDY=3THENK=A:GOTO280
4140 IFK=AANDSCRN(X,Y+A)=32THENGOSUB650
:GOTO500
4150 IFJ=-ATHENGOSUB600ELSEGOSUB620
4160 GOTO500
4170 PLOTX,Y,32:X=X+J:Y=Y+K:PLOTX,Y,41
4180 PLOT11,1,"^^":PLOT11,2,"_I":PLOT27
,1,"^^":PLOT27,2,"_I"
4185 FORN=0TO200STEP50:SOUNDA,N,15:WAIT
8:NEXT:SOUND1,0,15
4188 S(P)=S(P)+200
4190 GOSUB7000
4200 PLOT11,A,"?":PLOT11,2,"?":PLOT27
,A,"?":PLOT27,2,"?"
4210 GOSUB610
4220 PLOTX,Y,0:X=X+J:Y=Y+K:PLOTX,Y,41:G
OTO500
4230 S(P)=S(P)+INT(RND(1)*50)+A:GOSUB70
00
4235 FORN=0TO7:PLAYA,0,N,1000:NEXT:SOUN
D1,0,15
4250 IFJ*K<>0THEN4300
4260 IFJ=0ANDRND(A)>.5THENJ=-A
4265 IFJ=0THENJ=A
4270 IFK=0ANDRND(A)>.5THENK=A
4280 IFK=0THENK=-A
4290 GOTO500
4300 IFRND(A)>.5THENGOSUB650ELSEGOSUB64
0
4310 GOTO500
4320 PLOTX,Y,32:X=X+J:Y=Y+K
4330 PLOTX,Y,41
4331 FORI=ATOINT(RND(A)*3)+A
4332 B0=B0+300
4335 SOUNDA,300,15:WAITA:SOUNDA,400,15:
WAIT2:SOUNDA,600,15:WAIT5:SOUNDA,0,15
4340 M=S(P):N=P:P=5:S(P)=B0:GOSUB7000

```

```

4350 B0=S(P):P=N:S(P)=M
4355 NEXT I
4360 J=INT(RND(1)*3)-A:K=INT(RND(1)*3)-
A
4361 IFJ=0ANDK=-A THEN 4360
4365 S=S+10:W=0
4370 IFSCRN(X+J,Y+K)<>32 THEN 4360
4380 PLOTX,Y,79:X=X+J:Y=Y+K:PLOTX,Y,41
4390 GOTO 500
4400 IFJ*K<>0 THEN 4450
4410 IFJ=0ANDRND(A)>.5ANDSCRN(X+A,Y+K)=
32 THEN J=-A
4412 IFJ=0ANDSCRN(X-A,K+Y)=32 THEN J=A
4413 IFJ=0 THEN 4410
4414 IFK=0ANDRND(A)>.5ANDSCRN(X+J,Y-A)=
32 THEN K=-A
4415 IFK=0ANDSCRN(X+J,Y+A)=32 THEN K=A
4416 IFK=0 THEN 4414
4450 IFJ=-A THEN GOSUB 600:GOTO 500
4460 IFJ=A THEN GOSUB 620:GOTO 500
4470 IFJ=0 THEN J=A
4480 IFK=0 THEN K=-A
4490 IFJ=-AANDK=A THEN J=0
4492 IFJ=-AANDK=-A THEN J=A
4495 GOTO 500
4500 IFJ=0 THEN J=-A
4510 IFK=0 THEN K=-A
4520 IFJ=AANDK=A THEN J=-A
4522 IFJ=AANDK=-A THEN J=-A
4530 GOTO 500
4540 PLOTX,Y,32:X=X+J:Y=Y+K:PLOTX,Y,41
4550 PLAYA,0,4,5000:SOUNDA,350,15:SOUND
A,0,15
4555 IFB0>0 THEN PLOT1,23,"GREAT":PLOT3,2
4,"SHOT!"
4560 S(P)=S(P)+B0:GOSUB 7000
4565 IFB0>0 THEN FORU=AT035:SOUNDA,U*8,A5
:SOUND2,8,15:NEXT:SOUNDA,0,15
4570 J=A
4580 PLOTX,Y,93:X=X+J:Y=Y+K:PLOTX,Y,41
4585 PLOTA,23,"":PLOT3,24,""
4590 GOTO 500
4600 PLOTX,Y,32:X=X+J:Y=Y+K:PLOTX,Y,41

```

```

4604 PLAYA,0,4,5000:SOUNDA,350,15:SOUND
A,0,15
4606 IFB0>0 THEN PLOT1,23,"GREAT":PLOT3,2
4,"SHOT!"
4608 IFB0>0 THEN FORU=AT035:SOUNDA,U*8,A5
:SOUND2,8,15:NEXT:SOUNDA,0,15
4610 S(P)=S(P)+B0:GOSUB 7000
4620 J=-A
4630 PLOTX,Y,91:X=X+J:Y=Y+K:PLOTX,Y,41
4635 PLOTA,23,"":PLOT3,24,""
4640 GOTO 500
4650 H(A)=H(A)+A:IFH(A)=10 THEN PLOT31,22
,"2X":H(2)=2
4660 IFH(A)=20 THEN PLOT31,22," 3X":H(2
)=3
4670 IFH(A)=30 THEN PLOT31,22," 4X":
H(2)=4
4680 PLOTX,Y,32:X=X+J:Y=Y+K:PLOTX,Y,41
4690 J=-A:PLOTX,Y,123:X=X+J:Y=Y+K:GOTO 5
00
4700 H(A)=H(A)+A:IFH(A)=10 THEN PLOT31,22
,"2X":H(2)=2
4710 IFH(A)=20 THEN PLOT31,22," 3X":H(2
)=3
4720 IFH(A)=30 THEN PLOT31,22," 4X":
H(2)=4
4730 PLOTX,Y,32:X=X+J:Y=Y+K:PLOTX,Y,41
4740 J=A:PLOTX,Y,125:X=X+J:Y=Y+K:GOTO 50
0
7000 C=S(P)
7010 C=C/100000
7020 FORU=0 TO 5
7030 PLOT32+U,P*4,INT(C)+48
7040 C=C-INT(C)
7050 C=C*10
7060 NEXT U:RETURN
40000 FORN=46360 TO 46439:READA:POKEN,A:N
EXT
40010 FORN=46448 TO 46463:READA:POKEN,A:N
EXT
40020 FORN=46544 TO 46599:READA:POKEN,A:N
EXT
40030 FORN=46816 TO 46823:READA:POKEN,A:N
EXT

```

```

40040 FORN=46832T046847:READA:POKEN,A:N
EXT
40050 FORN=47072T047079:READA:POKEN,A:N
EXT
40055 A=1
40060 RETURN
40070 DATA0,0,0,0,1,30,32,31,0,0,1,14,4
8,0,15,48,0,12,50,13,13,2,60,0
40080 DATA0,12,19,44,44,16,15,0,0,0,32,
28,3,0,60,3,0,0,0,0,32,30,1,62
40085 DATA0,30,63,63,63,63,30,0
40090 DATA63,33,33,33,33,33,33,63,19,30
,50,30,19,30,50,30,19,30,50,18,63,30,12
40095 DATA0
40100 DATA63,61,58,52,40,16,32,32,32,32
,16,40,52,58,61,63,1,1,2,5,11,23,47,63
40110 DATA0,12,18,33,45,45,45,45,45,45,
45,45,45,45,45,45,45,45,45,33,18
40120 DATA12,0
40130 DATA12,30,63,45,30,18,12,0,63,63,
63,63,63,63,63,63,63,47,23,11,5,2,1,1
40140 DATA0,0,4,4,7,5,19,26,0,0,18,23,5
0,18,38,46,45,15,15,7,4,8,16,16
40150 DATA26,58,58,48,16,8,4,4
50000 REM
50005 POKE616,1
50010 R=32:P$="":GOSUB60000
50020 P$=CHR$(#C)+"DEMONS DEN":R=10:GOS
UB60000
50025 P$=CHR$(129):R=-3:GOSUB60000
50030 P$=CHR$(146):R=-4:GOSUB60000
50040 PRINTSPC(8)"????????????????????"
"
50050 PRINTSPC(8)"??????.-@??????.-@?????"
"
50060 PRINTSPC(8)"?????. @???. @????"
"
50070 PRINTSPC(8)"?????. @???"
"
50080 PRINTSPC(8)"??? > > > @??"
"
50090 PRINTSPC(8)"???. @?"
"
50100 PRINTSPC(8)"?. ; ; ; ; ?"
"

```

```

50110 PRINTSPC(8)"? = = = = ; ?"
"
50120 PRINTSPC(8)"?0 < ?"
"
50130 PRINTSPC(8)"?/? :< ?"
"
50140 PRINTSPC(8)"?? ;; ;; ?< ?"
"
50150 PRINTSPC(8)"?? ?= =? ?< ?"
"
50160 PRINTSPC(8)"?? == == ?< ?"
"
50170 PRINTSPC(8)"??? [?< ?"
"
50180 PRINTSPC(8)"??? {?< ?"
"
50190 PRINTSPC(8)"?? ?< ?"
"
50200 PRINTSPC(8)"?. @< ?"
"
50210 PRINTSPC(8)"? < ?"
"
50220 PRINTSPC(8)"?/? :< ?"
"
50230 PRINTSPC(8)"?/? :?< ?"
"
50240 PRINTSPC(8)"???/? :???< ?"
"
50250 PRINTSPC(8)"????/? :????< ?"
"
50260 PRINTSPC(8)"?????/? :????< ?"
"
50270 PRINTSPC(8)"?????? ?????< ?"
"
50280 PRINTSPC(8)"?????? ?????<?*?"
"
50290 PRINTSPC(8)"????????????????????+?"
"
50300 PRINTSPC(27)"", ";
50305 INK5:PAPER0
50310 RETURN
60000 FORI=1TOLEN(P$):POKE#BB83+R+I,ASC
(MID$(P$,I,1)):NEXTI:RETURN

```


CHARACTER REDEFINER

```

100 REM *****
110 REM *** Character Generator ***
120 REM *****
130 REM
140 REM By Kristjon Magnusson
150 REM
160 CLS:LORES0
170 PRINTCHR$(17)
180 X$=CHR$(27)
190 PRINT" ";CHR$(11);X$;"S";X$;"A
    INSTRUCTIONS ";X$;"P"
200 PRINT
210 PRINT" To determine the coordinate
where"
220 PRINT"you want to turn a cell 'off' or
'on'"
230 PRINT"you use the cursors. You use '0'
and"
240 PRINT"'1' to change the contents of ea
ch"
250 PRINT"cell. You now have three options
:":PRINT
260 PRINT" 1) Continue without selectin
g an enlarged character."
270 PRINT" 2) Entering a character from
the keyboard."
280 PRINT" 3) Entering the ASCII code o
f a character."
285 PRINTCHR$(11);CHR$(11);CHR$(11);CHR$(1
1);CHR$(11);CHR$(11)
290 FORI=0TO7
300 PLOT14,I+15,I+48
310 NEXT
320 FORI=0TO5
330 PLOTI+16,24,53-I
340 NEXT
350 FORN=15TO22
360 FORM=16TO21

```

```

370 PLOTM,N,20
380 NEXTM,N
390 GETOP$
400 IFOP$="1"OROP$="2"OROP$="3"THEN420
410 GOTO390
420 A$="
"
430 PLOT6,6,A$
440 FORR=1TO6
450 PLOT5,R+7,A$
460 NEXTR
470 IFOP$="1"THENK=32:GOTO500
480 IFOP$="2"THENINPUT" Enter the cha
racter ";K$:K=ASC(K$)
490 IFOP$="3"THENINPUT" Enter the ASC
II code ";K
500 N=46080+(8*K)
510 FORI=NTON+7
520 IFPEEK(I)>63THENPOKEI,PEEK(I)-64
530 NEXTI
540 FORI=0TO7:D=64:C=PEEK(N+I)
550 FORJ=0TO5:D=D-(D/2)
560 IFC>=DTHENC=C-D:K(I,J)=1
570 NEXTJ,I
580 FORI=0TO7
590 FORJ=0TO5
600 IFK(I,J)=1THENPLOT16+J,15+I,17ELSEPLOT
16+J,15+I,20
610 NEXTJ,I
620 PLOT2,25,"Press 'RETURN' when you are
finished"
630 PLOT14,15,176:PLOT16,24,181:Y=15:X=16
640 GETH$
650 IFH$=""THENPLOTX,Y,20
660 IFH$="1"THENPLOTX,Y,17
670 IFH$=CHR$(13)THEN810
680 IFH$=CHR$(10)THENY=Y+1:GOTO740
690 IFH$=CHR$(11)THENY=Y-1:GOTO730

```

```

700 IFH$=CHR$(8) THEN X=X-1:GOTO750
710 IFH$=CHR$(9) THEN X=X+1:GOTO760
720 GOTO640
730 IFY<15 THEN Y=15:GOTO640 ELSE 780
740 IFY>22 THEN Y=22:GOTO640 ELSE 770
750 IFX<16 THEN X=16:GOTO640 ELSE 790
760 IFX>21 THEN X=21:GOTO640 ELSE 800
770 PLOT14,Y-1,SCRN(14,Y-1)-128:PLOT14,Y,S
CRN(14,Y)+128:GOTO640
780 PLOT14,Y+1,SCRN(14,Y+1)-128:PLOT14,Y,S
CRN(14,Y)+128:GOTO640
790 PLOTX+1,24,SCRN(X+1,24)-128:PLOTX,24,S
CRN(X,24)+128:GOTO640
800 PLOTX-1,24,SCRN(X-1,24)-128:PLOTX,24,S
CRN(X,24)+128:GOTO640
810 FORM=0 TO 7
820 FORN=0 TO 5
830 P=SCRN(N+16,M+15)
835 C(N,M)=P
840 IFC(N,M)=17 THEN V(M)=V(M)+32/(2^N)
850 NEXTN
860 V(M)=V(M)+64

```

```

870 NEXTM
880 CLS:PAPER4:INK6:PRINT
890 PRINT"You can now either enter the,"P
RINT
900 PRINT"      1) Character,
      or"
910 PRINT"      2) ASCII code.":PRINT:PRINT
920 GETR$
930 IFR$="1" THEN INPUT"Enter the character
";F$:Q=ASC(F$):GOTO970
940 IFR$="2" THEN INPUT"Enter the ASCII code
";Q
950 IFQ<32 OR Q>126 THEN PRINT CHR$(11);CHR$(1
4)CHR$(11):GOTO940:ELSE 970
960 GOTO920
970 FORM=0 TO 7:POKE46080+(8*Q)+M,V(M):NEXTM
975 PRINT:PRINT:PRINT"  Press 'Y' if you w
ant to define      another character."
980 GETY$
990 IFY$="Y" THEN PRINT CHR$(17):RUN
995 PAPER7:INK0:CLS:PRINT CHR$(17)

```

3D ROTATION

```

50 REM=====
60 REM PROGRAM TO ROTATE ANY 3D SOLID
62 REM BODY ABOUT ANY AXIS PASSING
64 REM THROUGH ORIGIN THROUGH
66 REM PRESCRIBED STEPS,CHECK FOR
68 REM VISIBLE FACES,AND PRODUCE
70 REM CORRECT PERSPECTIVE PLOT.
72 REM=====
80 PAPER0:INK7:CLS:PRINT
94 PRINT CHR$(27);"J";SPC(2);"----- 3D
ROTATION -----"
95 PRINT CHR$(27);"J";SPC(2);"----- 3D

```

```

ROTATION -----":PRINT:PRINT:PRINT:PRINT
:PRINT:PRINT
96 PRINT CHR$(27);"J";"ORIGINAL PROGRAM BY
D.F.ROUND 12/2/82"
97 PRINT CHR$(27);"J";"ORIGINAL PROGRAM BY
D.F.ROUND 12/2/82":PRINT:PRINT
98 PRINT CHR$(27);"J";"ORIC-1 CONVERSION BY
M.J.BUNCE 5/1/84"
99 PRINT CHR$(27);"J";"ORIC-1 CONVERSION BY
M.J.BUNCE 5/1/84"
100 DIM D(6,4,2),R0(3,8),R1(3,8),A(3,3),VE(
3),VN(3),EY(3),VA(3),VB(3)

```

```

102 WAIT1000
104 REM=====
105 REM DATA I/P IN CORRECT ORDER
106 REM=====
110 FORV=1TO8
120 READR0(1,V),R0(2,V),R0(3,V)
130 NEXTV
131 DATA-25,-25,25,25,-25,25,25,-25,-25,-2
5,-25,-25
132 DATA-25,25,25,25,25,25,25,25,-25,-25,2
5,-25
140 FORF=1TO6
150 FORE=1TO4
160 READD(F,E,1),D(F,E,2)
170 NEXTE:NEXTF
171 DATA1,2,2,3,3,4,4,1
172 DATA5,1,1,4,4,8,8,5
173 DATA6,5,5,8,8,7,7,6
174 DATA2,6,6,7,7,3,3,2
175 DATA1,5,5,6,6,2,2,1
176 DATA3,7,7,8,8,4,4,3
178 CLS:PRINT:PRINT:PRINT:PRINT
180 INPUT"EYE POS'N.(APPROX. 150-200)";EY(
3)
190 EY(1)=0
200 EY(2)=0
210 INPUT"D.C'S.OF AXIS";U1,U2,U3
220 INPUT"TOTAL,STEP ROT'N.";TM,TT
230 T=0
240 MO=SQR(U1*U1+U2*U2+U3*U3)
250 U1=U1/MO
260 U2=U2/MO
270 U3=U3/MO
279 REM=====
280 REM TRANSFORMATION (ROTATION)
281 REM=====
290 SI=SIN(T/57.296)
300 CO=COS(T/57.296)
310 A(1,1)=U1*U1+CO*(1-U1*U1)
320 A(1,2)=U1*U2*(1-CO)-U3*SI
330 A(1,3)=U3*U1*(1-CO)+U2*SI
340 A(2,1)=U1*U2*(1-CO)+U3*SI
350 A(2,2)=U2*U2+CO*(1-U2*U2)
360 A(2,3)=U2*U3*(1-CO)-U1*SI
370 A(3,1)=U3*U1*(1-CO)-U2*SI
380 A(3,2)=U2*U3*(1-CO)+U1*SI
390 A(3,3)=U3*U3+CO*(1-U3*U3)
400 FORV=1TO8
410 FORI=1TO3
420 R1(I,V)=0
430 FORJ=1TO3
440 R1(I,V)=R1(I,V)+A(I,J)*R0(J,V)
450 NEXTJ:NEXTI:NEXTV
454 REM=====
455 REM CHECK FOR HIDDEN FACES
456 REM=====
460 CLS:HIRES
470 FORF=1TO6
472 FORJ=1TO3
474 VA(J)=R1(J,D(F,1,2))-R1(J,D(F,1,1))
476 VB(J)=R1(J,D(F,2,1))-R1(J,D(F,2,2))
478 NEXTJ
480 VN(1)=VA(2)*VB(3)-VA(3)*VB(2)
482 VN(2)=VA(3)*VB(1)-VA(1)*VB(3)
484 VN(3)=VA(1)*VB(2)-VA(2)*VB(1)
490 FORU=1TO3
500 VE(U)=EY(U)-R1(U,D(F,1,2))
510 NEXT U
520 DP=0
530 FORU=1TO3
540 DP=DP+VN(U)*VE(U)
550 NEXTU
560 IFDP<0THEN690
564 REM=====
565 REM PERSPECTIVE PROJECTION
566 REM=====
570 FORE=1TO4
580 V1=D(F,E,1)
590 PE=ABS(EY(3)/(R1(3,V1)-EY(3)))
600 X=INT(PE*R1(1,V1)+120)
610 Y=INT(PE*R1(2,V1)+100)
614 IFX<0THENX=0

```

```

615 IFX>239THENX=239
616 IFY>199THENY=199
617 IFY<0THENY=0
620 CURSETX,Y,3
630 V2=D(F,E,2)
640 PE=ABS(EY(3)/(R1(3,V2)-EY(3)))
650 N=INT(PE*R1(1,V2)+120)
660 M=INT(PE*R1(2,V2)+100)
665 B=N-X:C=M-Y

```

```

670 DRAWB,C,1
680 NEXTE
690 NEXTF
700 T=T+TT
710 IFT<=TMTHEN290
720 PRINT"ROTATION COMPLETE - PRESS ANY KE
Y ":GETA$
730 TEXT
740 END

```

SPELLBUZZ

```

~
10 REM"SPELLBUZZ"
15 T$="SPELLBUZZ"
20 FORA=47056TO47087:READB:POKEA,B:NEXT
30 DATA0,24,0,0,24,12,7,0,0,6,0,0,6,12,56,
0
40 DATA0,24,0,0,7,12,24,0,0,6,0,0,56,12,6,
0
50 Y$="z{":N$="|}"
55 DIMA$(100,3)
60 FORA=1TO100:FORB=1TO3:READA$(A,B):NEXTB
,A
100 REM DICTIONARY
101 DATA"PIECE","PEICE","PEECE"
102 DATA"SATURDAY","SATERDAY","SATTERDAY"
103 DATA"TUESDAY","TEUSDAY","TOOSDAY"
104 DATA"BANANA","BANNANA","BANANNA"
105 DATA"THREE","THEREE","THREA"
106 DATA"YELLOW","YELOW","YELLOU"
107 DATA"ALPHABET","ALFABET","ALFABIT"
108 DATA"FRIDAY","FRYDAY","FRIDEY"
109 DATA"COMPUTER","COMPUTOR","COMPUTAR"
110 DATA"OLIVER","OLIVAR","OLLIVER"
111 DATA"TELEVISION","TELIVISUN","TELYVISH
UN"
112 DATA"FOUR","FOOR","FUOR"
113 DATA"SUSANNAH","SUSANAH","SUSANNA"
114 DATA"SUPER","SOOPER","SOUPA"
115 DATA"THEIR","THIER","THEER"

```

```

116 DATA"EACH","EECH","EATCH"
117 DATA"HEAVY","HEVVY","EAVVY"
118 DATA"LEAST","LEEST","LAEST"
119 DATA"AGAIN","AGEN","AGIN"
120 DATA"CUDDLE","CUDDLE"<"CUDEL"
121 DATA"THANKS","FANKS","THANGKS"
122 DATA"CONNECT","CONECT","CONNEKT"
123 DATA"INTRODUCTION","INTREDUCTION","INT
ERODUCTION"
124 DATA"AVERAGE","AVARAGE","AVIRAGE"
125 DATA"FIRST","FERST","FIRSST"
126 DATA"GIRL","GERL","GURL"
127 DATA"TABLE","TABEL","TABIL"
128 DATA"LOOP","LOUP","LUPE"
129 DATA"MOUTH","MOUF","MOWTH"
130 DATA"PARALLEL","PARALEL","PARALELL"
131 DATA"PERFORM","PURFORM","PIRFORM"
132 DATA"PURSUE","PERSUE","PIRSUE"
133 DATA"PERSUADE","PURSUADE","PERSWADE"
134 DATA"STATEMENT","STATMENT","STATEMEANT
"
135 DATA"REQUIRE","REKWIRE","REQUIR"
136 DATA"LETTER","LETER","LETTAR"
137 DATA"LETTUCE","LETTICE","LETTIS"
138 DATA"TELEPHONE","TELEFONE","TELIFONE"
139 DATA"ENEMY","ENIMY","ENEMIE"
140 DATA"MOUSE","MOWSE","MOWESE"
141 DATA"TRIANGLE","TRYANGLE","TRIANGEL"

```


142 DATA"COMPLETE", "COMPLEET", "COMPLEATE"
 143 DATA"COMMA", "COMMAR", "COMMARE"
 144 DATA"THINK", "FINK", "THYNK"
 145 DATA"MUSIC", "MUSIK", "MUESIK"
 146 DATA"WHICH", "WICH", "WHICHE"
 147 DATA"TOGETHER", "TOGEVVER", "TWOGEVVER"
 148 DATA"THEME", "THEEM", "FEEM"
 149 DATA"REVEAL", "REVEEL", "RE-VEAL"
 150 DATA"WONDERFUL", "WUNDERFUL", "WONDERFUL"

L"

151 DATA"BEGINNING", "BEGINING", "BEEGINNING"

"

152 DATA"BEAUTY", "BAEUTY", "BEUATY"
 153 DATA"ECHO", "ECHOE", "ECKOE"
 154 DATA"CHOIR", "KWIRE", "KWOIR"
 155 DATA"TRUMPET", "TRUMPETTE", "TRUMPETT"
 156 DATA"HEAT", "HEET", "HEATE"
 157 DATA"PROMISE", "PROMISS", "PROMIS"
 158 DATA"WORLD", "WIRLD", "WORLED"
 159 DATA"MINUTE", "MINETTE", "MINIT"
 160 DATA"WHEEL", "WEELL", "WEILL"
 161 DATA"HEIGHT", "HITE", "HYTE"
 162 DATA"FAITHFUL", "FAYTHFULL", "FAITHFULL"
 163 DATA"THOUGHT", "THOUGHT", "THORT"
 164 DATA"DESCEND", "DEESEND", "DESEND"
 165 DATA"NECESSARY", "NECCESARY", "NECESSAR"

Y"

166 DATA"MOUNTAIN", "MOWNTAIN", "MOWNTEN"
 167 DATA"DIFFERENCE", "DIFERENCE", "DIFFERRE"

NCE"

168 DATA"VALLEY", "VALLY", "VALEY"
 169 DATA"HEART", "HAERT", "HARTE"
 170 DATA"EIGHT", "AITE", "EAGHT"
 171 DATA"SEVEN", "SEVVEN", "SEVIN"
 172 DATA"WEDNESDAY", "WENSDAY", "WENESDAY"
 173 DATA"JANUARY", "JANWARY", "JANURY"
 174 DATA"FEBRUARY", "FEBWARY", "FEBUARY"
 175 DATA"AUGUST", "AWGUST", "ORGUST"
 176 DATA"SEPTEMBER", "SETEMBER", "SEPTEMBAR"
 177 DATA"DECEMBER", "DESEMBER", "DICEMBER"
 178 DATA"MALICE", "MALISS", "MALIS"

179 DATA"LAUGH", "LAFF", "LARF"
 180 DATA"DOCTOR", "DOCTER", "DOCTAR"
 181 DATA"REASON", "REESON", "RIESON"
 182 DATA"CREATURE", "KWEETURE", "CREETURE"
 183 DATA"BECAUSE", "BECOS", "BEECAUSE"
 184 DATA"GREATEST", "GRATEST", "GREATIST"
 185 DATA"WHISPER", "WISPER", "WISSPER"
 186 DATA"TROUBLE", "TRUBBLE", "TROUBEL"
 187 DATA"INFERENCE", "INFERENCE", "INFERANC"

E"

188 DATA"PEOPLE", "PEEPEL", "PEEPIL"
 189 DATA"TURNED", "TIRNED", "TURNED"
 190 DATA"SPECIAL", "SPESHILL", "SPESHALL"
 191 DATA"VOICE", "VOYCE", "VOISE"
 192 DATA"BUILDING", "BILLDING", "BILDING"
 193 DATA"MOTHER", "MUVVER", "MOTHAR"
 194 DATA"GEM", "GEMM", "JEM"
 195 DATA"JEWELLERY", "JEWELRY", "JEWELREY"
 196 DATA"CRYSTAL", "CRISTAL", "CRYSTEL"
 197 DATA"DIFFICULTY", "DIFICULTY", "DIFFIKUL"

TY"

198 DATA"SWORD", "SORD", "SWORED"
 199 DATA"ISSUE", "ISYOU", "ISUE"
 200 DATA"ACCOMMODATION", "ACOMODATION", "ACC

OMODATION"

250 POKE524,255
 260 DOKE48002,3099
 270 FORA=1TO9
 280 POKE48014+A,ASC(MID\$(T\$,A,1))
 290 NEXT
 295 DOKE48024,2075
 300 CLS
 310 PRINTSPC(200)SPC(230)CHR\$(4)CHR\$(27)"N
 "T\$CHR\$(4)
 320 PRINT:PRINT:PRINTSPC(10)CHR\$(96)" ROB
 SIBTHORPE"
 330 PRINT:PRINT:PRINTSPC(7)"Press any key
 to start"
 340 POKE#26A,2
 345 PING
 350 A\$=KEY\$

```

360 IFA$=""THEND=RND(1):D=FRE(""):GOTO350
370 POKE#26A,3
400 FORN=1TO10
405 CLS
410 D=INT(RND(1)*100)+1:GOSUB1200
415 IFFTHENF=0:GOTO410
420 A=INT(RND(1)*3)+1
430 B=INT(RND(1)*3)+1
440 IFB=ATHEN430
450 C=6-(A+B)
500 PRINT:PRINT:PRINT"Which spelling is co
rrect?"
510 PRINT:PRINT:PRINT,A$(D,A)
520 PRINT:PRINT,A$(D,B)
530 PRINT:PRINT,A$(D,C)
535 PRINT,"=====
540 PRINT:PRINTSPC(1)
542 PLOT3,24,"-----
--"
545 PLOT3,25,"Type answer - then press RET
URN"
550 INPUTB$
555 PRINT,"=====
565 WAIT100:PRINT
568 PRINTSPC(2)CHR$(4)CHR$(27)"J";
570 FORE=1TOLEN(A$(D,1))
580 PRINTMID$(A$(D,1),E,1);
585 PING:WAIT100
590 NEXT
600 PRINT " ";
615 IFB$=A$(D,1)THENPRINTY$:Y=Y+1
618 IFB$<>A$(D,1)THENPRINTN$
620 PRINTCHR$(4)
625 PRINT:PRINT:PRINT:PRINT,,,"You have be
en correct":PRINT,,,Y" out of "N
628 WAIT200
630 NEXT
700 CLS
705 POKE#26A,2

```

```

710 PRINTSPC(200)SPC(150)CHR$(4)CHR$(27)"N
";
720 IFY=10THENPRINTY$" Brilliant "Y$:GOSUB
800
730 IFY<10ANDY>6THENPRINT"Fairly good":GOS
UB900
740 IFY<7ANDY>3THENPRINTN$" Oh Dear! "N$:G
OSUB1000
750 IFY<4THENPRINTN$,N$,N$:GOSUB1100
755 PRINTCHR$(4)
758 Y=0:N=0
760 WAIT200:GOTO300
800 FORX=5000TO100STEP-100:SOUND1,X,12:WAI
T10:NEXT:EXPLODE:RETURN
810 S=SIN(X)*100
815 IFS<0THENS=-S
820 SOUND1,S,9
830 WAIT5
840 NEXT
900 FORX=1TO50
910 S=SIN(X)*100
915 IFS<0THENS=-S
920 SOUND1,S,9
930 WAIT5
940 NEXT
950 FORX=10TO15:SOUND1,S,X:WAIT5:NEXT
960 FORX=14TO0STEP-1:SOUND1,S,X:WAIT5:NEXT
970 RETURN
1000 FORX=50TO1STEP-1:SOUND1,X*100,9:WAIT5
:NEXT:WAIT50:SOUND1,0,0:RETURN
1100 FORX=1TO10
1110 SOUND1,5000,12:WAIT20:SOUND1,0,0:WAIT
10:NEXT:RETURN
1120 RETURN
1200 D(N)=D
1210 FORX=N-1TO1STEP-1
1220 IFD(N)=D(X)THENF=-1:X=1
1230 NEXT
1240 RETURN

```

ANALYSIS

```

10 REM"ANALYSIS"
20 GRAB
30 CLS
40 PRINT:POKE#26A,2
50 PRINTSPC(255)SPC(133)CHR$(4)CHR$(27)"N
ANALYSIS":PRINTCHR$(4)
60 PLOT7,15,CHR$(96)+" ROB SIBTHORPE 1983"
70 PLOT2,24,"Press any key to start progra
mme"
80 GET A$
90 POKE#26A,3
100 CLS
105 PRINT:PRINT"WHAT IS THE TITLE OF THIS
ANALYSIS?"
110 INPUTTA$
115 IFLEN(TA$)>16THENTAS=LEFT$(TA$,12)+"..
"
120 PRINT:PRINT:PRINT,CHR$(4)CHR$(27)"J"TA
$:PRINTCHR$(4)
130 INPUT"HOW MANY ENTRIES WILL THERE BE";
TE
135 DIMZ$(TE)
140 PRINT:PRINT"Please type column heading
s required. Make COST the last one"
145 PRINT:PRINT"MAXIMUM 10 HEADINGS"
150 REPEAT
155 INPUTA$
160 GOSUB200000
165 PRINT"OK"
170 UNTILA$="COST"
175 WAIT50
200 CLS
210 PLOT6,10,"NOW FOR THE ENTRIES"
220 PLOT6,12,"Press any key to start"
230 GETA$
240 CLS
250 FORA=1TOTE

```

```

260 PRINT:PRINT"ENTRY "A
270 PRINTB$;TAB(25-LEN(B$));
280 INPUTB$(A)
290 IFC$=""THEN560
300 PRINTC$;TAB(25-LEN(C$));
310 INPUTC$(A)
320 IFD$=""THEN560
330 PRINTD$;TAB(25-LEN(D$));
340 INPUTD$(A)
350 IFE$=""THEN560
360 PRINTE$;TAB(25-LEN(E$));
370 INPUTE$(A)
380 IFF$=""THEN560
390 PRINTF$;TAB(25-LEN(F$));
400 INPUTF$(A)
410 IFG$=""THEN560
420 PRINTG$;TAB(25-LEN(G$));
430 INPUTG$(A)
440 IFH$=""THEN560
450 PRINTH$;TAB(25-LEN(H$));
460 INPUTH$(A)
470 IFI$=""THEN560
480 PRINTI$;TAB(25-LEN(I$));
490 INPUTI$(A)
500 IFJ$=""THEN560
510 PRINTJ$;TAB(25-LEN(J$));
520 INPUTJ$(A)
530 IFK$=""THEN560
540 PRINTK$;TAB(25-LEN(K$));
550 INPUTK$(A)
560 NEXT
570 WAIT50
1000 CLS
1010 PRINTSPC(40)CHR$(4)CHR$(27)"N"TA$:PRI
NTCHR$(4)
1020 PRINT:PRINT:PRINT"Which of the follow
ing do you require?"

```

```

1030 PRINT:PRINT"1 EXTRACT COSTS"
1040 PRINT"2 LIST PARTICULAR ENTRY"
1050 PRINT"3 LIST COLUMN VARIATIONS"
1060 PRINT"4 LIST LIKE ENTRIES"
1070 PRINT"5 LIST ALL ENTRIES"
1080 PRINT"6 LIST SPARE ITEMS"
1090 PRINT"7 CHANGE ENTRY"
1100 PRINT"8 STOP
1120 PRINT:PRINT"Press number"
1130 GETA$
1140 V=VAL(A$)
1150 CLS
1160 ONVGO SUB2000,3000,4000,5000,6000,7000
,8000,10000
1170 GOTO1000
2000 PRINT:PRINT:PRINT"EXTRACT COSTS"
2010 PRINT:PRINT"INDICATE WHICH CATEGORIES
YOU WANT UNDER EACH COLUMN"
2020 PRINT:PRINT"...or ALL if appropriate"
:PRINT:PRINT
2030 IFB$="COST"THEN2300
2040 PRINTB$;TAB(25-LEN(B$));
2050 INPUTBX$
2060 IFC$="COST"THEN2300
2070 PRINTC$;TAB(25-LEN(C$));
2080 INPUTCX$
2090 IFD$="COST"THEN2300
2100 PRINTD$;TAB(25-LEN(D$));
2110 INPUTDX$
2120 IFE$="COST"THEN2300
2130 PRINTE$;TAB(25-LEN(E$));
2140 INPUTEX$
2150 IFF$="COST"THEN2300
2160 PRINTF$;TAB(25-LEN(F$));
2170 INPUTFX$
2180 IFG$="COST"THEN2300
2190 PRINTG$;TAB(25-LEN(G$));
2200 INPUTGX$
2210 IFH$="COST"THEN2300
2220 PRINTH$;TAB(25-LEN(H$));
2230 INPUTHX$

```

```

2240 IFI$="COST"THEN2300
2250 PRINTI$;TAB(25-LEN(I$));
2260 INPUTIX$
2270 IFJ$="COST"THEN2300
2280 PRINTJ$;TAB(25-LEN(J$));
2290 INPUTJX$
2300 PRINT:PRINT,CHR$(27)"L HANG ON!"
2310 WAIT50
2320 C3=0:C4=0:C5=0
2330 FORA=1TOTE
2340 C1=0:C2=0
2350 IFB$="COST"ANDC1=C2THENC3=VAL(B$(A)):
C4=C4+C3:C5=C5+1:GOTO2820
2360 IFB$(A)="0"THEN2820
2370 C1=C1+1
2380 IFB$(A)=BX$ORBX$="ALL"THENC2=C2+1
2390 IFC1<>C2THEN2820
2400 IFC$=""THEN2820
2410 IFC$="COST"ANDC1=C2THENC3=VAL(C$(A)):
C4=C4+C3:C5=C5+1:GOTO2820
2420 C1=C1+1
2430 IFC$(A)=CX$ORCX$="ALL"THENC2=C2+1
2440 IFC1<>C2THEN2820
2450 IFD$=""THEN2820
2460 IFD$="COST"ANDC1=C2THENC3=VAL(D$(A)):
C4=C4+C3:C5=C5+1:GOTO2820
2470 C1=C1+1
2480 IFD$(A)=DX$ORDX$="ALL"THENC2=C2+1
2490 IFC1<>C2THEN2820
2500 IFE$=""THEN2820
2510 IFE$="COST"ANDC1=C2THENC3=VAL(E$(A)):
C4=C4+C3:C5=C5+1:GOTO2820
2520 C1=C1+1
2530 IFE$(A)=EX$OREX$="ALL"THENC2=C2+1
2540 IFC1<>C2THEN2820
2550 IFF$=""THEN2820
2560 IFF$="COST"ANDC1=C2THENC3=VAL(F$(A)):
C4=C4+C3:C5=C5+1:GOTO2820
2570 C1=C1+1
2580 IFF$(A)=FX$ORFX$="ALL"THENC2=C2+1
2590 IFC1<>C2THEN2820

```



```

2600 IFG$="" THEN 2820
2610 IFG$="COST" AND C1+C2 THEN C3=VAL(G$(A)):
C4=C4+C3:C5=C5+1:GOTO 2820
2620 C1=C1+1
2630 IFG$(A)=GX$ OR GX$="ALL" THEN C2=C2+1
2640 IF C1<>C2 THEN 2820
2650 IFH$="" THEN 2820
2660 IFH$="COST" AND C1=C2 THEN C3=VAL(H$(A)):
C4=C4+C3:C5=C5+1:GOTO 2820
2670 C1=C1+1
2680 IFH$(A)=HX$ OR HX$="ALL" THEN C2=C2+1
2690 IF C1<>C2 THEN 2820
2700 IFI$="" THEN 2820
2710 IFI$="COST" AND C1=C2 THEN C3=VAL(I$(A)):
C4=C4+C3:C5=C5+1:GOTO 2820
2720 C1=C1+1
2730 IFI$(A)=IX$ OR IX$="ALL" THEN C2=C2+1
2740 IF C1<>C2 THEN 2820
2750 IFJ$="" THEN 2820
2760 IFJ$="COST" AND C1=C2 THEN C3=VAL(J$(A)):
C4=C4+C3:C5=C5+1:GOTO 2820
2770 C1=C1+1
2780 IFJ$(A)=JX$ OR JX$="ALL" THEN C2=C2+1
2790 IF C1<>C2 THEN 2820
2810 C3=VAL(K$(A)):C4=C4+C3:C5=C5+1
2820 NEXT
2830 CLS
2840 PRINT:PRINT B$;TAB(25-LEN(B$));BX$
2850 PRINT C$;TAB(25-LEN(C$));CX$
2860 PRINT D$;TAB(25-LEN(D$));DX$
2870 PRINT E$;TAB(25-LEN(E$));EX$
2880 PRINT F$;TAB(25-LEN(F$));FX$
2890 PRINT G$;TAB(25-LEN(G$));GX$
2900 PRINT H$;TAB(25-LEN(H$));HX$
2910 PRINT I$;TAB(25-LEN(I$));IX$
2920 PRINT J$;TAB(25-LEN(J$));JX$
2930 PRINT K$;TAB(25-LEN(K$));KX$
2940 PRINT:PRINT "C4" FOR "C5" ITEMS"
2950 PRINT:PRINT "Any more EXTRACT COSTS re
quired?";
2960 GETA$

```

```

2970 IFA$="Y" THEN 2000
2980 RETURN
3000 PRINT:PRINT:PRINT "PARTICULAR ENTRY"
3010 PRINT:PRINT "Which one";
3020 INPUTA
3030 IFA>T THEN PRINT "TOO HIGH - CHOOSE FRO
M 1 TO "TE:GOTO 3020
3040 GOSUB 22000
3050 PRINT:PRINT "Do you want another PARTI
CULAR ENTRY?";
3060 GETA$
3070 IFA$="Y" THEN 3000
3080 RETURN
4000 PRINT:PRINT:PRINT "COLUMN VARIATIONS"
4005 FORA=1 TO T:Z$(A)="" :NEXT
4010 PRINT:INPUT "Which column";Z$
4020 GOSUB 23000
4030 PRINT:PRINT Z$(1)
4040 FORA=2 TO T
4050 FL=0
4060 FORN=A-1 TO 0 STEP -1
4070 IF Z$(N)=Z$(A) THEN FL=1:N=N-1
4080 NEXTN
4090 IF FL=0 THEN PRINT Z$(A)
4100 NEXTA
4110 PRINT:PRINT "Do you want more COLUMN V
ARIATIONS?";
4120 GETA$
4130 IFA$="Y" THEN 4000
4140 RETURN
5000 PRINT:PRINT:PRINT "LIKE ENTRIES"
5010 PRINT:INPUT "Which column";Z$
5020 PRINT:INPUT "Which variation";ZX$
5030 GOSUB 23000
5040 FORA=1 TO T
5050 IF Z$(A)=ZX$ THEN GOSUB 22000
5055 WAIT 200
5060 NEXT
5070 PRINT:PRINT "Do you want other LIKE EN
TRIES?";
5080 GETA$

```

```

5090 IFA$="Y"THEN5000
5100 RETURN
6000 PRINT:PRINT:PRINT"LIST ALL ENTRIES":P
RINT
6010 FORA=1TOTE
6030 GOSUB22000
6040 WAIT200
6045 NEXT
6050 PRINT:PRINT"...again?"
6060 GETA$
6070 IFA$="Y"THEN6000
6080 RETURN
7000 PRINT:PRINT:PRINT"SPARE ITEMS"
7010 FORA=1TOTE
7020 IFB$(A)="0"THENPRINTA,
7030 NEXT
7040 PLOT2,20,"Press any key to clear scre
en"
7050 GETA$
7060 RETURN
8000 PRINT:PRINT:PRINT"CHANGE ENTRY"
8010 PRINT:INPUT"Which one";A
8020 PRINT:PRINTB$;TAB(25-LEN(B$));B$(A)
8050 GOSUB24000
8060 B$(A)=Z$(A)
8070 IFC$=""THEN8430
8080 PRINTC$;TAB(25-LEN(C$));C$(A)
8090 GOSUB24000
8100 C$(A)=Z$(A)
8110 IFD$=""THEN8430
8120 PRINTD$;TAB(25-LEN(D$));D$(A)
8130 GOSUB24000
8140 D$(A)=Z$(A)
8150 IFE$=""THEN8430
8160 PRINTE$;TAB(25-LEN(E$));E$(A)
8170 GOSUB24000
8180 E$(A)=Z$(A)
8190 IFF$=""THEN8430
8200 PRINTF$;TAB(25-LEN(F$));F$(A)
8210 GOSUB24000
8220 F$(A)=Z$(A)

```

```

8230 IFG$=""THEN8430
8240 PRINTG$;TAB(25-LEN(G$));G$(A)
8250 GOSUB24000
8260 G$(A)=Z$(A)
8270 IFH$=""THEN8430
8280 PRINTH$;TAB(25-LEN(H$));H$(A)
8290 GOSUB24000
8300 H$(A)=Z$(A)
8310 IFI$=""THEN8430
8320 PRINTI$;TAB(25-LEN(I$));I$(A)
8330 GOSUB24000
8340 I$(A)=Z$(A)
8350 IFJ$=""THEN8430
8360 PRINTJ$;TAB(25-LEN(J$));J$(A)
8370 GOSUB24000
8380 J$(A)=Z$(A)
8390 IFK$=""THEN8430
8400 PRINTK$;TAB(25-LEN(K$));K$(A)
8410 GOSUB24000
8420 K$(A)=Z$(A)
8430 WAIT50
8440 CLS
8450 PRINT:PRINT:PRINT"ENTRY "A" NOW READS
:"
8460 GOSUB22000
8470 PRINT:PRINT"Do you want to CHANGE ano
ther ENTRY?";
8480 GETA$
8490 IFA$="Y"THEN8000
8500 RETURN
10000 PRINTCHR$(17)
10010 PRINTSPC(154)CHR$(4)CHR$(27)"N ANALY
SIS : "TA$
10020 PRINTCHR$(4)
10030 PLOT5,15,"To continue, type CONT"
10040 PLOT5,17,".....and press RETURN"
10050 STOP
10060 PRINTCHR$(17)
10070 RETURN
20000 FL=FL+1
20010 IFFL=1THENB$=A$:DIMB$(TE)

```

```

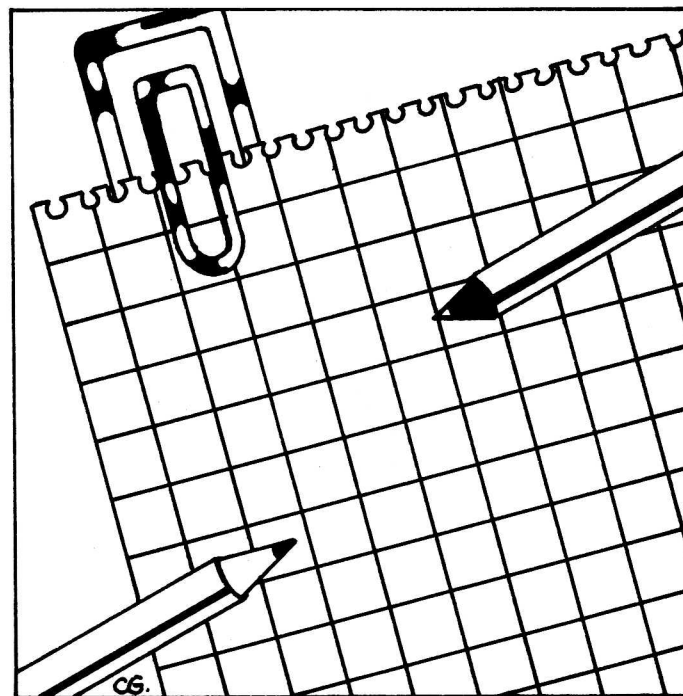
20020 IFFL=2THENC$=A$:DIMC$(TE)
20030 IFFL=3THEND$=A$:DIMD$(TE)
20040 IFFL=4THENE$=A$:DIME$(TE)
20050 IFFL=5THENF$=A$:DIMF$(TE)
20060 IFFL=6THENG$=A$:DIMG$(TE)
20070 IFFL=7THENH$=A$:DIMH$(TE)
20080 IFFL=8THENI$=A$:DIMI$(TE)
20090 IFFL=9THENJ$=A$:DIMJ$(TE)
20100 IFFL=10THENK$=A$:DIMK$(TE)
20110 RETURN
22000 PRINT"ENTRY "A:PRINT
22005 PRINTB$;TAB(25-LEN(B$));B$(A)
22010 IFC$=""THEN22190
22020 PRINTC$;TAB(25-LEN(C$));C$(A)
22030 IFD$=""THEN22190
22040 PRINTD$;TAB(25-LEN(D$));D$(A)
22050 IFE$=""THEN22190
22060 PRINTE$;TAB(25-LEN(E$));E$(A)
22070 IFF$=""THEN22190
22080 PRINTF$;TAB(25-LEN(F$));F$(A)
22090 IFG$=""THEN22190
22100 PRINTG$;TAB(25-LEN(G$));G$(A)
22110 IFH$=""THEN22190
22120 PRINTH$;TAB(25-LEN(H$));H$(A)
22130 IFI$=""THEN22190
22140 PRINTI$;TAB(25-LEN(I$));I$(A)
22150 IFJ$=""THEN22190
22160 PRINTJ$;TAB(25-LEN(J$));J$(A)
22170 IFK$=""THEN22190
22180 PRINTK$;TAB(25-LEN(K$));K$(A)
22190 PRINT
22200 RETURN
23000 FORA=1TOTE
23010 IFZ$=B$THENZ$(A)=B$(A):GOTO23110
23020 IFZ$=C$THENZ$(A)=C$(A):GOTO23110
23030 IFZ$=D$THENZ$(A)=D$(A):GOTO23110
23040 IFZ$=E$THENZ$(A)=E$(A):GOTO23110
23050 IFZ$=F$THENZ$(A)=F$(A):GOTO23110
23060 IFZ$=G$THENZ$(A)=G$(A):GOTO23110
23070 IFZ$=H$THENZ$(A)=H$(A):GOTO23110
23080 IFZ$=I$THENZ$(A)=I$(A):GOTO23110

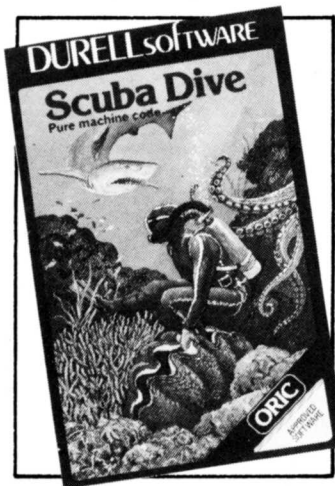
```

```

23090 IFZ$=J$THENZ$(A)=J$(A):GOTO23110
23100 IFZ$=K$THENZ$(A)=K$(A):GOTO23110
23110 NEXT
23120 RETURN
24000 PRINT"...now to read ";
24020 INPUTZ$(A)
24030 PRINT"OK
24040 RETURN

```





SCUBA DIVE

Durell Software
£6.95

Oric 16K & 48K

I must admit it, I have never been a fan of Durell's 'Harrier Attack', I have always felt that they could do better. 'Scuba Dive' is a far more original and interesting game. You have to get your diver to jump off a boat and then swim around underwater trying to avoid sharks, octopi and other denizens of the deep. Your task is to pick up pearls from oysters and collect minerals and gold from undersea caverns. Of course, you have a time limit controlled by the amount of oxygen in your tanks.

Graphically, the game is good although not quite as good as the version I saw on the Spectrum and lots of fun.

LOCHNESS MONSTER

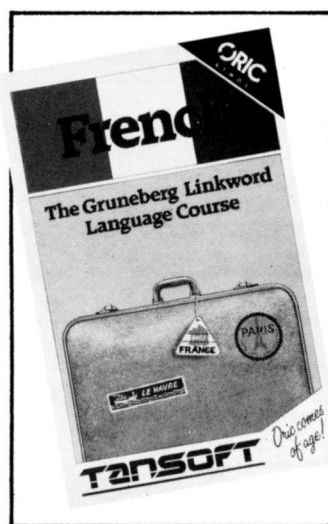
Romik Software
£6.99

Oric 48K

Romik Software are well established in the Sinclair area and have ventured into Oric Software with Lochness Monsters. To the strains of 'Loch Lomond' you pilot your boat around Lochness trying to pick up fish. When you have collected 25 you are awarded a poison fish which can be used as bait for the monsters which appear from time to time.

You also have harpoons which you can hurl at the monsters but these will only kill them if they hit the head.

Although the movement of the graphics is rather jerky, the scrolling of the lake is quite impressive as you move your boat around the screen. Altogether a good entrance into the Oric fielded by an already well known company.



GRUNEBERG LINKWORD COURSE

Tansoft Limited
£12.95

Oric-1 & Atmos 48K

Available for French, Italian, German and Spanish, these

courses give a good basic introduction to these languages. You are supplied with an instruction manual, a program cassette and a spoken vocabulary cassette all in a video cassette sized box. The courses are based on a method created by Professor Michael Gruneberg of the University of Swansea. Basically, the method is based on word association. For instance, the Spanish word for 'Lobster' is 'Humma'. You are told to imagine a lobster with a sense of humour, thus immediately giving you something to picture in your mind and an aid to later recall. The silliness of some of the phrases actually helps you to remember the words.

You are taught around 400 words and a good background in grammar. These courses are ideal for those who are going abroad for holidays or business and can give you a lot of confidence if you are just beginning to learn a language.

THE ULTRA

P.S.S.

£6.95

Oric 48K

Every so often a piece of software is produced which becomes a standard for a particular computer. Where the Oric-1 is concerned the first game to fall into that category was, undoubtedly, Xenon 1. The second, I believe, is The Ultra.

The idea behind The Ultra cannot in any way be described as new, the basic principle being to vaporise as many aliens as possible before they do the same to you. The cassette sleeve describes The Ultra as a pure arcade game and this is certainly the case.

There are 16 different screens of aliens, each having their own shape and flight formation. Part of the addictive quality of this game

lies in trying to discover what the next screen of aliens look like.

Screen one of Ultra 1 consists of three rows of green craft which move across the screen from right to left, snapping their jaws as they go. I was soon able to negotiate these without losing any of my five lives. Screen 2 turned out to be a bank of menacing blue space invaders. I soon discovered that shooting the craft from the left of the formation created a 'safe' area on the screen where I could position my spaceship without any danger of being bombed. A similar principle applies to many of the other screens: in fact the main skills of The Ultra lie in knowing where to position the spaceship at the beginning of each screen and in adoption the safest and most efficient method of tackling the opposition. I won't give the game away by telling you how to play each screen, except to say that Ultra 16 is very hard and the reward for completing it will have been well earned.

Just in case 'The Ultra' sounds too easy, one of the game's features is to penalise indiscriminate use of the fire button by informing you that your laser has overheated.

When this happens you just have to wait until it cools down before you can continue firing.

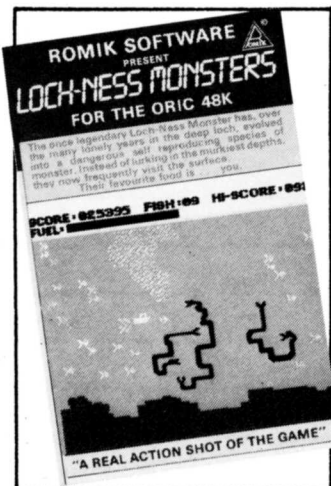
I would rate 'The Ultra' as one of the best games of its type. If you are going to buy only a few games for your Oric, buy this one.

BOZY BOA

CDS Microsystems
£5.95

Oric 48K

This game is a variation of the old Snakes game. Basically you have to control your Boa around the screen avoiding the side and eating beetles and snails and avoiding the flowers which glow over the screen.



Software Scan

People have been heard to complain that there is not much software available for the Oric-1. This listing, incomplete as it is, should show that that statement is patently untrue. We have over 160 programs on our files, and that's more than enough for most owners.

Of course this listing is far from complete.

If you know of any software that we are not listing, then please contact us at Oric Owner and we will add the new programs to our listing. We shall endeavour to be as comprehensive as possible, though there is no guarantee

that all the software listed will be available at time of going to press, and there may well be additional items from some of the companies.

A growing number of software houses are making their software Oric-1 and Atmos compatible by recording two versions of the

programs on the cassettes. If you are an Atmos owner, check with the company or your dealer as to whether the software is suitable.

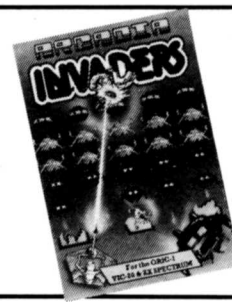
You will also have to check whether the software is suitable for a 16K machine. Much of it is for the 48K version only.

Program	Type	Price	Distributors
3D Invaders	Arcade Game	£6.95	Quark Data
3D Maze/Breakout Game		£7.50	IJK
Ace in the Hole	Adventure	£5.95	Wild Electronics
Acheron's Rage	Arcade	£6.95	Softek
Airline	Strategy	£6.95	CCS
Arcaine Quest	Adventure	£5.95	Wild Electronics
Assembler	Utility	£9.95	Superior
Assembler/Diss.	Utility	£6.95	Durell
Assembler/Editor	Utility	£7.95	QED
Asteroids	Arcade	£5.95	Arctic
Author	Business	£14.50	Tansoft
Awari	Board	£5.50	Kenema
Battleships	Game	£6.90	Aztec
Beseiged	Education	£9.95	Sulis
Bozy Boa	Game	£5.95	CDS



Bridge	Board	£9.85	ICB
Candyfloss	Game	£7.50	IJK
Cassette 50	Game	£9.95	Cascade
Centipede	Arcade	£6.95	SS
Champions	Strategy	£6.95	Peaksoft
Chess	Board	£9.99	Tansoft
Chicken	Arcade	£6.95	Durell
Click	Game	£4.95	Headfield

Program	Type	Price	Distributors
Composer	Utility	£6.50	Sector 7
Dallas	Strategy	£6.95	CCS
Database	Business	£19.95	Gemini
Death Satellite	Adventure	£6.90	A & F Software
Defence Force	Arcade	£7.95	Tansoft
Design Aid	Utility	£5.50	Kenema
Dinky Kong	Arcade	£6.95	Severn



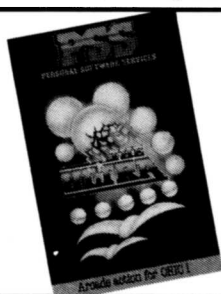
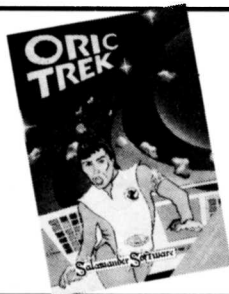
Dracula	Adventure	£6.90	Mr Micro
Draculas Revenge	Arcade	£6.95	Softek
Elementary Maths	Education	£6.90	Aztec
Evictor	Arcade	£7.95	Wizard
Extension Mon.	Utility	£15.00	Kenema
Fantasy Quest	Adventure	£6.50	IJK
Filestar	Business	£12.00	Kenema
Fishy Business	Adventure	£9.95	Salamander
Flight	Simulation	£4.50	Tansoft
Flight Simulator	Simulation	£6.95	Quark Data
Forth	Utility	£15.00	Tansoft
Franklin's Tomb	Adventure	£9.95	Salamander
Galactic Defend	Arcade	£4.95	Headfield
Galaxians	Arcade	£6.95	Softek
Galaxy 5	Game	£6.95	Durell
Games 01	Game	£7.95	Salamander

Software Scan

Program	Type	Price	Distributors
Gamespack	Game	£7.00	Sector 7
Gauntlet	Arcade	£6.95	PSS
Grail	Adventure	£6.95	Severn



Gremlin	Arcade	£6.95	Wintersoft
Harrier Attack	Arcade	£6.95	Durell
Hells Temple	Adventure	£12.00	Kenema
Home Management	Business	£10.95	QED
Hopper	Arcade	£6.95	PSS
Horror Atoll	Adventure	£5.95	Wild Electronics
House of Death	Adventure	£9.99	Tansoft
Hunchback	Arcade	£6.90	Ocean
Ice Giant	Arcade	£6.95	Softtek
Invaders	Arcade	£7.50	IJK
Invaders	Arcade	£6.95	PSS
Invaders	Arcade	£5.50	Arcadia
Invaders	Arcade	£6.99	Prog. Factory
Island of Death	Game	£6.90	Ocean
Jerico 2	Adventure	£6.95	Elephant



Jogger	Arcade	£6.95	Severn
Johnny Reb	Strategy	£6.95	Lothlorien
Jungle Trouble	Game	£6.95	Durell
Just a Mot	Education	£9.95	Sulis
Key Trainer	Business	£5.50	Kenema

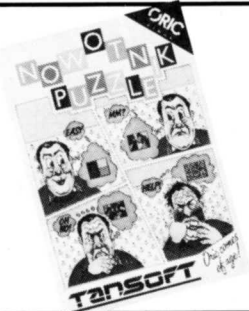
Program	Type	Price	Distributors
Killer Caverns	Adventure	£5.95	Virgin
Kindersoft	Education	£9.95	Softshop
Leopard Lord	Adventure	£5.95	Wild Electronics
Light Cycle	Arcade	£6.95	PSS
Lightning Ass.	Utility	£9.90	Mr Micro
Lochness Monster	Arcade	£6.99	Romik
Loki	Game	£6.45	Joe the Lion
Lone Raider	Arcade	£6.95	Severn
Lords of Time	Adventure	£9.90	Level 9
Lost in Space	Adventure	£9.95	Salamander
Lunar Mission	Game	£7.95	Superior
Mastermynde	Game	£4.95	Headfield
Maths 4	Education	£5.90	IMS
Mines of Saturn	Adventure	£5.95	Mikro-Gen



Moonster	Arcade	£6.95	Quark Data
Moria	Adventure	£6.95	Severn
Mr Wimpy	Game	£6.90	Ocean
Multigames 1	Game	£6.90	Tansoft
Multigames 2	Game	£6.90	Tansoft
Mushroom Mania	Arcade	£5.50	Arcadia
Mutant Invaders	Arcade	£5.95	R & R
Nowotnik	Game	£6.90	Tansoft
Oric Golf	Game	£5.95	R & R
Oric Trek	Game	£9.95	Salamander
Oric-1 Blitz	Game	£4.95	Kompsoft
Oric-1 Composer	Utility	£4.95	Kompsoft
Oric-1 Forth	Utility	£6.95	Kompsoft
Oric-Cad	Utility	£9.99	Tansoft
Oric-Calc	Business	£14.50	Tansoft

Software Scan

Program	Type	Price	Distributors
Oric-Mon	Utility	£15.00	Tansoft
Oricade	Utility	£8.50	Severn
Oricbase	Business	£14.50	Tansoft
Oricle	Game	£6.95	Quark Data
Oricmon	Utility	£8.95	PSS
Oricmunch	Arcade	£7.95	Tansoft
Oricstar	Business	£12.00	Durell
Orion Ass./Diss.	Utility	£12.95	Lothlorien
Paragram	Game	£6.00	Elephant
Pasta Blasta	Arcade	£5.50	Arcadia
Personal Finance	Business	£9.99	Juniper
QED Toolkit	Utility	£5.95	QED
Quizmaster 1/2	Educational	£5.59	R & R
Rat Splat	Arcade	£7.95	Tansoft
Records	Business	£10.00	Peach



Reverse	Board	£6.50	IJK
Reversi	Board	£5.95	CDS
Ring of Darkness	Adventure	£9.95	Wintersoft
Ringo	Game	£6.50	Elephant
Road Frog	Arcade	£6.90	Ocean
Roundesy	Adventure	£5.95	Wild Electronics
Runelord	Adventure	£7.50	Modular Concept
Scuba Dive	Game	£6.95	Durell
Snowball	Adventure	£9.95	Level 9
Space Crystal	Adventure	£6.95	Quark Data
Space Docker	Game	£4.95	Kompssoft
Space Shuttle	Game	£8.00	Microdeal
Spread Sheet	Business	£19.95	Gemini
Squirm	Arcade	£6.90	Aztec

Star Bingo	Game	£5.95	R & R
Starfighter	Arcade	£6.95	Durell
Starship	Arcade	£6.50	Sector 7
Super Breakout	Arcade	£4.50	Tansoft
Super Fruit	Game	£6.95	Quark Data
Super Meteors	Arcade	£6.95	Softek
Super Trek	Game	£6.95	Mogul
T-Comp	Utility	£12.95	Bamby
Terminator	Arcade	£6.50	Impact
Terror . . . Deep	Adventure	£5.95	Wild Electronics
Tevrog's Kingdom	Adventure	£9.99	Cable
The Hobbit	Adventure	£14.95	Tansoft M. House
The Ultra	Arcade	£6.95	PSS
Toady	Arcade	£4.95	Headfield
Toolkit	Utility	£9.95	Superior
Toolkit	Utility	£8.95	IMS
Treasure House	Game	£6.99	Prog. Factory
Two Gun Turtle	Arcade	£6.95	Lothlorien
Ultima Zone	Arcade	£8.50	Tansoft
Vanquisher	Game	£6.50	Elephant
Variety Pack 25	Game	£5.95	QED
Video Challenge	Game	£6.95	QED
Warlord	Strategy	£6.95	Lothlorien
Waydor	Adventure	£7.50	IMS
Word Processor	Business	£17.25	Juniper
Wordpower	Education	£9.95	Sulis
World Geography	Educational	£7.95	Superior
Xenon-1	Arcade	£8.50	IJK
Zodiac	Adventure	£9.99	Tansoft
Zodiac	Adventure	£6.90	A & F Software
Zorgons Revenge	Arcade	£8.50	IJK
Backgammon	Game	£7.95	Dormere
Orical Invaders	Arcade	£5.95	Dormere
Oricaid	Utility	£11.95	Micrograf
M.A.R.C.	Arcade	£6.95	PSS

SPRECHEN SIE DEUTSCH?

No, but this English/German quiz from David White of Malvern, Worcestershire, may help you.

The letters a,o,u and b are poked at the beginning to give the German letters with umlaut accents. Use these letters in your answers.

Oric will print phrases and you'll have to supply the translation. The dictionary can easily be expanded with the help of a German phrase book. Remember to increase N in line 10 if you extend the dictionary.

```
5 REM ** NUMBER OF WORD PAIRS **
10 N=12
15 REM ** POKE a,o,u,b **
20 FORI=1TO4:READA:GOSUB260:NEXTI
50 DATA46856,34,28,34,34,62,34,34,0
60 DATA46968,34,28,34,34,34,34,28,0
70 DATA47016,34,0,34,34,34,34,30,0
80 DATA46864,20,26,18,22,18,17,30,32
90 DIME$(N):DIMD$(N)
100 FORI=1TON:READE$(I):READD$(I):NEXTI
165 REM ** RANDOM SELECTION **
170 K=INT(RND(1)*N+1)
180 J=INT(RND(1)+.5)
190 IFJ=0THEN230
195 REM ** PRINTING ENGLISH **
200 PRINTE$(K):INPUTA$
210 IFA$=D$(K)THENPRINT"DAS IST RICHTIG
?"ELSEPRINT"NEIN?" "+D$(K)
220 GOTO170
225 REM ** PRINTING GERMAN **
230 PRINTD$(K):INPUTA$
240 IFA$=E$(K)THENPRINT"CORRECT?"ELSEPR
INT"NO?" "+E$(K)
250 GOTO170
255 REM ** CHARACTER POKE ROUTINE **
260 FORJ=ATO A+7:READD:POKEJ,D:NEXTJ:RET
URN
300 DATA THE HOUSE,DAS HAUS,THE DOG,DER
HUND,I SEE, ICH SEHE,HELLO,GUTEN TAG
310 DATA GOODBYE,AUF WIEDERSEHEN,TO SLE
EP,SCHLAFEN,OH,ACH SO
320 DATA MERRY CHRISTMAS,FROLICHE WEINA
CHTEN,I DON'T KNOW,ICH WEISS NICHT
330 DATA OVER NINETY,UBER NEUNZIG,UGLY,
HAESSLICH,ANGRY,BOESE
```

SOUND DEMO

Here's a short program from David Peat that demonstrates the sound capabilities of the Oric. Pressing S stops the noise. Is it music, we ask?

```
10 CLS:PAPER4:INK7:PRINTCHR$(6)
20 REPEAT
30 :SOUND INT(RND(1)*4)+1,INT(RND(1)*700
)+10,0
40 :PLAY7,7,INT(RND(1)*7)+1,INT(RND(1)*500
)+50:WAITRND(1)*100
50 UNTILFALSEORKEY$="S":PLAY0,0,0,0
60 END
```

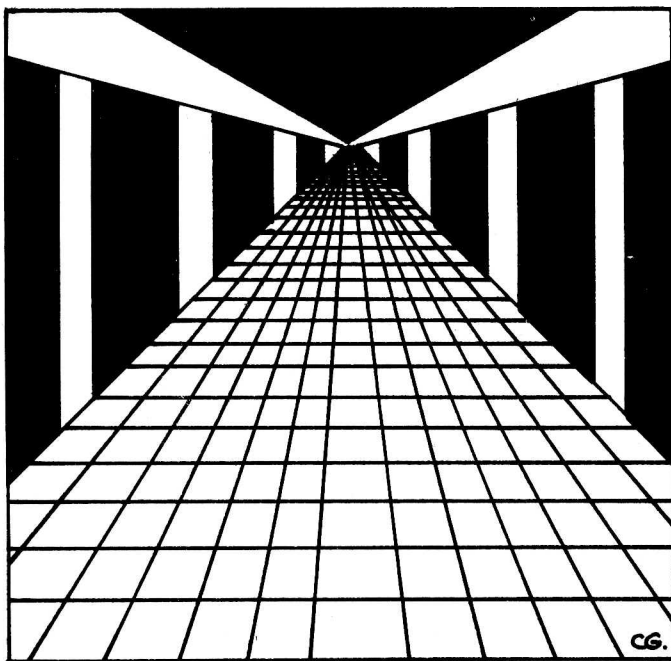
REMEMBER THE NUMBER

Some of us are better at remembering numbers than others. This Quickie from Mr Wood of Cheadle, Cheshire, tests that out.

It displays a seven digit number for a brief time and then prompts you to enter what you thought the number was. Changing the WAIT time in line 100 will change the display time.

```
10 CLS
20 PRINT
30 PRINT"REMEMBER THE NUMBER"
40 WAIT100
50 CLS
60 TARG=INT(RND(1)*9999999)+1
70 IFTARG<10000000THEN60
80 PRINT:PRINT:PRINT:PRINT
90 PRINTTARG
100 WAIT150
110 CLS
120 PRINT:PRINT:PRINT:PRINT
130 PRINT"WHAT IS THE NUMBER?"
140 PRINT
150 INPUTTRY
160 PRINT
170 IFTRY=TARGTHENPRINT"CORRECT"
180 IFTRY<>TARGTHENPRINT"WRONG"
190 PRINT
200 PRINT"ANOTHER TRY (Y/N)?"
210 INPUTAN$
220 PRINT
230 IFAN$="Y"THEN30
240 IFAN$="N"THENEND
```


DRUM PRACTICE



While your Oric may not be a Buddy Rich or Ringo Starr, this short program from J. A. McNamara of Wigan will demonstrate the drum machine potential of the Oric. Put the sound through an amplifier and you'll be surprised at how good it is.

```
0 REM***DRUMS BY J.A.McNAMARA***
10 PRINTCHR$(6)
20 GOTO150
30 MUSIC1,1,1,0:PLAY1,0,1,80
40 WAITRND(1)*100+50
50 RETURN
60 MUSIC2,2,2,0:PLAY0,2,1,200
70 WAITRND(1)*20
80 RETURN
90 MUSIC3,6,6,0:PLAY4,0,1,60
100 WAITRND(1)*20:GOSUB60
110 RETURN
120 MUSIC1,3,3,0:PLAY1,0,1,80
130 WAITRND(1)*50:GOSUB60
140 RETURN
150 A=RND(1)*4+1.5
160 ON AGOSUB 30,60,80,120
170 GOTO150
```

ELLIPSES

An ellipse is a regular oval. This Quickie from twelve year old Martin Williams of Newmarket will draw out five random ellipses on the HIRES screen.

To give more ellipses, change the 5 in line 15. To make the ellipses smoother, use numbers higher than 200 in line 30. Remember that the more steps there are, the longer the program will take to execute.

```
1 REM COPYRIGHT M.WILLIAMS
10 HIRES
15 FORP=1TO5
20 A=INT(RND(1)*50):B=INT(RND(1)*50)
30 FORT=1TO200
40 CURSET120+COS(T)*B,100+SIN(T)*A,1
50 NEXTT
60 NEXTP
```

SONG FOR CHILDREN

```
10 REM *** SONG FOR CHILDREN ***
20 FORN=1TO50
30 READK,T,W
40 W=W*.4
50 MUSIC1,K,T,0
60 PLAY1,0,1,5000
70 WAITW
80 NEXTN
90 RESTORE:WAIT300:GOTO20
100 DATA 2,10,12,3,3,37,3,2,12,3,3,
,37,3,5,12
110 DATA 3,6,37,3,5,12,3,3,37,3,2,
12,3,3,50
120 DATA 3,3,50,3,3,87
130 DATA 2,10,12,3,3,37,3,2,12,3,3,
,37,3,5,12
140 DATA 3,6,37,3,5,12,3,3,37,3,2,
12,3,3,50
150 DATA 3,3,50,3,3,87
160 DATA 3,6,12,3,5,37,3,4,12,3,5,
37,3,6,12
170 DATA 3,8,50,3,5,37,3,5,12,3,6,
37,3,5,12
180 DATA 3,6,37,3,10,12,4,3,50,3,1
0,37,3,10,12
190 DATA 3,11,37,3,10,12,3,8,37,3,
6,12,3,5,37
200 DATA 3,3,12,3,2,37,3,,12,3,3,
50,3,3,50
210 DATA 3,3,87
```

Martin Hagström

UNION JACK

The Oric-1 is made in Britain. here's a short program from K. J. Diver that results in a full colour display of the Union Jack, using the graphics mode.

The programs works by the simple expedient of CURSET and FILL commands. It starts off slowly but finishes with a flourish and doesn't take long to run. Ther's a repeat option at the end too. Now, do any of our overseas readers have listings that will print out the flags of other countries?

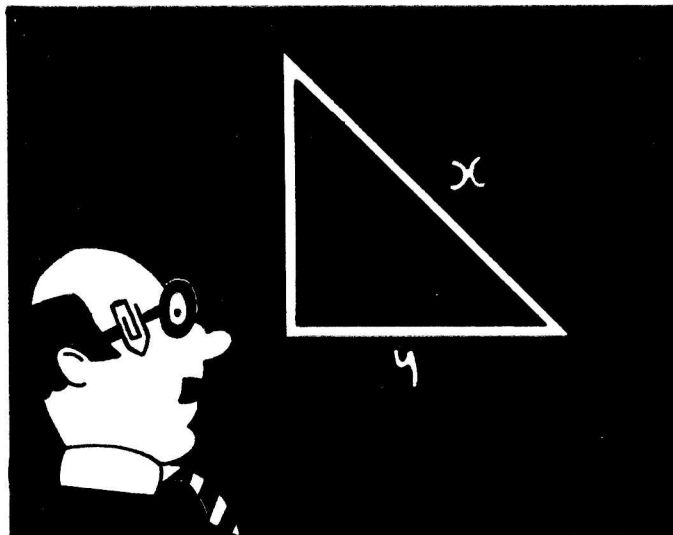
```

5 REM ***UNION JACK***
10 PAPER0:INK7:HIRES:PRINTCHR$(17
)
20 A=20:B=1:C=48:D=232:E=1:F=45:G
OSUB1000
30 A=150:B=-5:C=42:D=228:E=-1:F=-
45:X=0:Y=0:Z=0:GOSUB1000
40 CURSET222,20,0:FILL131,3,64
50 CURSET12,28,0:FILL112,1,4
60 CURSET135,20,0:FILL41,1,4
70 CURSET135,110,0:FILL41,1,4
80 CURSET105,20,0:FILL131,1,1
90 CURSET111,20,1:FILL130,4,127
100 CURSET18,71,1:FILL29,34,127
110 CURSET12,71,0:FILL29,1,1
120 CURSET0,0,0:FILL170,1,23
200 INPUT"FLY THE FLAG AGAIN (Y/N
)";A$
210 IFA$="Y"THENRUN
220 IFA$="N"THENTEXT:GOTO230
225 GOTO200
230 PAPER7:INK0:END
1000 REPEAT
1010 CURSET0,A+Y,0:FILL1,1,4
1020 CURSET6,A+Y,1:DRAWB+X,0,1
1030 CURSET12+X,A+Y,0:FILL1,1,1
1040 CURSET18+X,A+Y,1:DRAW18,0,1
1050 CURSETC+X,A+Y,0:FILL1,1,4
1060 CURSETC+X+6,A+Y,1:DRAW138-2*
X,0,1
1070 CURSET198+Z,A+Y,0:FILL1,1,1
1080 CURSET204+Z,A+Y,1:DRAW18,0,1

1090 CURSETD-X,A+Y,0:FILL1,1,4
1100 CURSETD-X+6,A+Y,1:DRAW1+X,0,
1
1110 X=X+1.5:Y=Y+E:Z=Z-1.5
1120 UNTILY=F
1130 RETURN
  
```

BLACKBOARD

Martin Walker is a teacher and uses this Quickie to display characters double height when they are entered at the keyboard.



```

10 REM *** BLACKBOARD ***
20 PRINTCHR$(20)
30 CLS
40 POKE618,10
50 FORI=0TO26
60 PLOT1,I,10
70 NEXTI
80 K=1
90 J=2
100 GETA$
110 A=ASC(A$)
120 IFA<32ORA>126THEN180
130 PLOTJ,K,A
140 PLOTJ,K+1,A
150 J=J+1
160 IFJ=39THEN90
170 GOTO100
180 IFA=3THEN270
190 IFA=13THEN250
200 IFA=27THEN30
210 IFA=127ANDJ>2THENJ=J-1
220 PLOTJ,K,32
230 PLOTJ,K+1,32
240 GOTO100
250 K=K+4
260 IFK<26THEN100
263 GOTO30
270 PRINTCHR$(4);CHR$(20)
280 POKE618,3
290 CLS
300 END
  
```

and forthly . . . ROLL

ROLL is standard in Forth-79. Executing 'n ROLL' will rotate the n'th number down the stack to the top of the stack.

So, $n_1 n_2 n_3 n_4 n_5$ becomes

$n_2 n_3 n_4 n_5 n_1$

'3 ROLL' is the same as ROT and '2 ROLL' equates to SWAP

ROLL pushes the n-1 topmost entries on the Data

Stack (DS) to the Return Stack (RS). All <R and R< items in the listing are stored as within a loop, the loop limits are above RS. At this stage, the target nth number is on top of DS. All the numbers on RS are then pulled back to DS, one by one, and swapped with the nth number. If $n = 1$, ROLL has no effect.

```
: ROLL DUP 1 = IF DROP ELSE DUP 1 DO SWAP R> R> ROT
  >R >R >R LOOP 1 DO R> R> ROT ROT >R >R
  SWAP LOOP THEN;
```

ARRAY

ARRAY defines arrays of any dimension, and uses ROLL. You can define as many array fields as you wish.

To build a one field array, FRED, with five integer cells, enter 5 1 ARRAY FRED.

To dimension a two field array of four by ten element cells, called MICK, enter 4 10 2 ARRAY MICK.

The general form for an N dimensional array of t_1 by t_2 by... t_N cells, is $t_1 t_2 \dots t_N$ N ARRAY NAME

Executing 2 FRED returns the address of element 2 in the array FRED. Executing 2 6 MICK puts the address of element 2,6 on top of the stack.

You'll note that the range of element numbers includes 0 as the first element. So, FRED elements are indexed by 0,1,2,3,4, and MICK by 0-3, 0-9.

Note too that no range checks are carried out. Enter an element that doesn't exist, such 7 FRED, -1 FRED or 2 10 MICK and you run the risk of reading or writing in the definitions of other words.

ARRAY works like this. The >BUILDS part stores N in the next free cell, loops N times, storing the number (t) in the next free cell and multiplies it by the others. The final product is then multiplied by

two to give the number of bytes needed for the array data. This number is then ALLOTTed. The memory map (Diag. 1) shows what is happening.

The DOES part is a little more complicated. An entry ' $x_1 x_2 x_3 \dots x_N$ NAME' should get the address (AD2) of element number $x_1 x_2 x_3 \dots x_N$ pushed to the top of DS. The address ADO, containing N, is pushed to the top of DS when DOES is executed. The offset (OF1) from here to the first address (AD1), containing data, is $2*(N+1)$ bytes.

If $n=1$ then the offset (OF2) from AD1 to AD2 is x_1 . If $N=2$ then OF2 equals $x_1 + x_2 * t_1$, and if $n=3$, the OF2 equals $x_1 + t_1(x_2 + t_2 * x_3)$. The general formula for OF2 is $OF2 = x_1 + t_1(x_2 + t_2 * (... (x_{N-1} + t_{N-1} * x_N) ...))$. The offsets have been measured in integer cells. To get the byte offset, just multiply by two.

So, $AD2 = ADO + 2*(OF1 + OF2)$.

DOES thus gets N from memory and checks if $N=1$. If so, $OF1=4$ (bytes) and $OF2=2*x_1$ (bytes). Therefore, $AD2=ADO+4+2*x_1$ bytes. If $N=1$, $OF1=2*(N+1)$. OF2 is calculated from the OF2 general formula, starting from the innermost parenthesis.

The program loops $n-1$

times, each time multiplying the result by the next t and adding the next x. This result

is multiplied by 2 to give the byte offset, and this total is added to both OF1 and ADO.

```
: ARRAY <BUILDS 1 SWAP DUP , 0 SWAP DO 1 1+ ROLL
  DUP , * -1 +LOOP 2 * ALLOT DOES> DUP @ DUP 1 = IF
  DROP 4 + SWAP ELSE 1 SWAP DO DUP 1 1 - 2 * + @
  ROT * ROT + SWAP -1 +LOOP DUP @ 1+ ROT + THEN
  2 * + ;
```

Memory Map		
ADDRESS	CONTENTS	COMMENTS
AD0	N t_1 t_2 . . . t_N	This address is pushed on stack when 'DOES' is executed.
AD1	(0,0,...,0,0) (1,0,...,0,0) . . . ($t_1-1,0,...,0,0$) (0,1,0,...,0,0) . . .	The content of the address above here are <i>not</i> amended when the array once is defined.
AD2	($x_1, x_2, \dots, x_N$) . . . ($t_1-1, t_2-1, \dots, t_N-1$)	Data part. Address of element no. ($x_1, x_2, \dots, x_N$)

Memory map - the parentheses mark that the content of the cell is an element of the array. Note that each line in the map is one integer cell or *two* bytes.

CLA

CLA, CLear Array, is used to clear any array. It uses the ERASE function, and the form is 'CLA NAME'. There is no need to enter the array dimensions. CLA consists of two words.

(CLA) assumes that the PFA of the array to be cleared is at the top of the DS. It multiplies all the ts in a loop that is executed N times. This number is then multiplied by two to give the number of bytes of memory that are to be ERASEd.

The start address used by ERASE is given by $ADO+2*(N+1)$. The address on the stack when -FIND is

executed is two bytes lower than ADO, as the first cell of the PFA points to the DOES> part of ARRAY. That's why there is a 2+ at the start of (CLA).

CLA tries to find the array. If not found, an error message is displayed and the routine QUITs. If the array is found, a check is made to see if the computer is in the compilation or execution state.

If execution, (CLA) is called. Otherwise, PFA is compiled as a literal and (CLA) called. Thus you can use CLA both directly from the keyboard and within another word definition.

```
: (CLA) 1 SWAP 2+ DUP @ 0 DO DUP 1 1+ 2 * + @ ROT
  SWAP LOOP DUP @ 2 * + 2+ SWAP 2 * ERASE;
: CLA -FIND IF DROP STATE @ IF [COMPILE] LITERAL
  COMPILE (CLA) ELSE (CLA) THEN ELSE 0 MESSAGE SP!
  QUIT THEN ; IMMEDIATE
```

APPEND/ RENUMBER

Ever wished that your collection of useful BASIC routines you got from magazines and suchlike could be rationalised into one single file? And how about the times that you wished that you could tag programs together instead of retyping something that you had done some months before?

With this machine code utility for the 48K model you could do precisely that; and as a bonus, you can also use it to renumber programs with a first line number and increment of your choosing. Three error and two OK messages are also provided so that any programs which fail to 'park' or append would not be accidentally overwritten or NEWed.

Loading the source code

Ideally, a purpose-made assembler or monitor should be used, but if one is not available, then the BASIC Loader/mini-monitor provided should do. If you are using an assembler, then substitute all RTS instructions with BRK and replace them when each routine has been tested and found to be satisfactory. The zero-page locations used are 70H to 7FH and the description of breakpoint values are given in the listing. Use a short BASIC program to do this; you will be able to recognize the end of the program by a series of three zero bytes.

Using the BASIC Loader/mini-monitor

Enter the Loader and simply RUN the lines you require. The input parameter will only accept hex values in the form of a string input; do not include '#' or spaces. Stop entry by entering 'S' in response to the prompt.

Test each routine by RUNning 100. Enter the address to execute in response to the prompt and the routine will be executed, displaying zero-page locations 70H to 7FH on RTS. Enter 'S' to quit; any other key will prompt you for another address to execute.

RUN 200 to examine the contents of memory. It will dump eight locations at a time and wait for a key press before displaying the next. Pressing 'S' will stop the routine.

CSAVE the completed program with:
CSAVE "APPEND", A#B200, E#B443, AUTO

I suggest that you use the Slow mode, but this depends

on your faith! To Load the program from tape, simply enter:

CLOAD "APPEND"

Usage instructions

Before entering or Loading any BASIC programs, you will need to load APPEND first. Then enter NEW and CLEAR or you will find that the computer will insist on displaying the "?OUT OF MEMORY" message. APPEND will not be affected. If the program does not auto-start on loading, enter CALL #B200 to bring up the installed message and to ensure that APPEND is in memory.

To park a BASIC program, enter:

CALL #B420

You should then get the "OK - IN MEMORY" message. Programs larger than 6.5K will generate a "?SIZE ERROR" and will not be parked. You should not attempt to enter HIRES at any time as this will erase APPEND and any parked programs.

To append the program, enter:

CALL #B430

Note that you cannot recall a parked program into 'empty' memory; at least one BASIC line must be present - i.e. it must be LISTable. After all, the object is to append a program! You may append a parked program as many times as you like as long as the combined programs do not exceed 37.75K in which case the "?APPEND SIZE ERROR" message will be generated. Both programs however will still be intact but will not be combined.

Line renumbering

APPEND will renumber all lines of the appended program using the last line number of the program it appended to as base reference. The original increments are kept; so for example if the last line numbered 4000 and the parked program had 10 as its first line, then the first line of the appended portion will be numbered 4010. If renumbering exceeds 65535 (FFFFH) the APPEND will attempt to renumber the entire program in increments of 10 starting with 10. If this still produces a line number greater than 65535, then the message "?RENUMBER ERROR" is displayed and the program will need to be renumbered 'manually'.

'Manual' renumbering

Enter:

DOKE #74, with the first line number of your choice.

DOKE #76, with the increment.

CALL #B319

Note that only the line numbers are renumbered, APPEND will not check for GOTO branches.

I can let any reader who has difficulty in entering the source code have a copy of the program on cassette at £4.00 each and would be pleased to try and assist anyone if they send an SAE to:
David Yan, 16 Durban Road, Smethwick, Warley, West Midlands B66 3SQ

B200: 4C EA B3 JMP \$B3EA

Find end of BASIC program

#70: LSB Last line
#71: MSB position
#72: LSB End of program
#73: MSB position

B203: A9 01 LDA #\$01
B205: 85 70 STA \$70

Renumber from appended portion

#70: LSB Last line position
#71: MSB
#72: LSB Current line pos.
#73: MSB
#74: Constant=last line no.
#75: MSB Constant
#76: LSB Increment
#77: MSB

B207: A9 05 LDA #\$05
B209: 85 71 STA \$71
B20B: A0 01 LDY #\$01
B20D: B1 70 LDA (\$70),Y
B20F: 99 72 00 STA \$0072,Y
B212: 88 DEY
B213: F0 F8 BEQ \$B20D
B215: C8 INY
B216: B1 72 LDA (\$72),Y
B218: F0 0A BEQ \$B224
B21A: A5 72 LDA \$72
B21C: 85 70 STA \$70
B21E: A5 73 LDA \$73
B220: 85 71 STA \$71
B222: D0 E7 BNE \$B20B
B224: C8 INY
B225: B1 72 LDA (\$72),Y
B227: D0 F1 BNE \$B21A
B229: 60 RTS
B22A: D8 CLD
B22B: 38 SEC
B22C: A5 73 LDA \$73
B22E: E9 05 SBC #\$05
B230: 48 PHA
B231: 38 SEC
B232: E9 1A SBC #\$1A
B234: B0 18 BCS \$B24E
B236: 68 PLA
B237: 85 7D STA \$7D

Check size and load Move values

#78: LSB From
#79: MSB
#7A: LSB To

#7B: MSB
#7C: Remaining no. of bytes
#7D: No. of pages to move

B239: A5 72 LDA \$72
B23B: 85 7C STA \$7C
B23D: A9 01 LDA #\$01
B23F: 85 78 STA \$78
B241: A9 05 LDA #\$05
B243: 85 79 STA \$79
B245: A9 01 LDA #\$01
B247: 85 7A STA \$7A
B249: A9 98 LDA #\$98
B24B: 85 7B STA \$7B
B24D: 60 RTS
B24E: 68 PLA
B24F: 60 RTS
B250: A6 7D LDX \$7D

Block move
on values contained in
#78-#7D

B252: F0 10 BEQ \$B264
B254: A0 00 LDY #\$00
B256: B1 78 LDA (\$78),Y
B258: 91 7A STA (\$7A),Y
B25A: 88 DEY
B25B: D0 F9 BNE \$B256
B25D: E6 79 INC \$79
B25F: E6 7B INC \$7B
B261: CA DEX
B262: D0 F0 BNE \$B254
B264: A4 7C LDY \$7C
B266: B1 78 LDA (\$78),Y
B268: 91 7A STA (\$7A),Y
B26A: 88 DEY
B26B: 10 F9 BPL \$B266
B26D: F0 F7 BEQ \$B266
B26F: 60 RTS
B270: D8 CLD

Check size of new program and
load new Move values

#78: LSB From
#79: MSB
#7A: LSB To
#7B: MSB
#7D: Remain
#7E: Pages

B271: 18 CLC
B272: A5 72 LDA \$72
B274: 65 7C ADC \$7C
B276: A5 73 LDA \$73
B278: 65 7D ADC \$7D
B27A: 38 SEC
B27B: E9 98 SBC #\$98


```

B27D: B0 10      BCS $B28F
B27F: A9 01      LDA #$01
B281: 85 78      STA $78
B283: A9 98      LDA #$98
B285: 85 79      STA $79
B287: A5 72      LDA $72
B289: 85 7A      STA $7A
B28B: A5 73      LDA $73
B28D: 85 7B      STA $7B
B28F: 60         RTS
B290: A5 72      LDA $72

```

Repoint BASIC line pointers

#78: LSB Current position
 #79: MSB being worked on
 #A: LSB Next position
 #7B: MSB to be updated
 #74: Value to subtract
 #75: Value to be subtracted
 #76: Result=no. of bytes in the line

```

B292: 85 7A      STA $7A
B294: A5 73      LDA $73
B296: 85 7B      STA $7B
B298: A9 01      LDA #$01
B29A: 85 74      STA $74
B29C: A0 01      LDY #$01
B29E: B1 7A      LDA ($7A),Y
B2A0: F0 32      BEQ $B2D4
B2A2: A5 7A      LDA $7A
B2A4: 85 78      STA $78
B2A6: A5 7B      LDA $7B
B2A8: 85 79      STA $79
B2AA: 88         DEY
B2AB: B1 78      LDA ($78),Y
B2AD: 85 75      STA $75
B2AF: D8         CLD
B2B0: 38         SEC
B2B1: A5 75      LDA $75
B2B3: E5 74      SBC $74
B2B5: 85 76      STA $76
B2B7: 18         CLC
B2B8: A5 78      LDA $78
B2BA: 65 76      ADC $76
B2BC: 85 7A      STA $7A
B2BE: A5 79      LDA $79
B2C0: 69 00      ADC #$00
B2C2: 85 7B      STA $7B
B2C4: A5 75      LDA $75
B2C6: 85 74      STA $74
B2C8: A0 01      LDY #$01
B2CA: B9 7A 00   LDA $007A,Y
B2CD: 91 78      STA ($78),Y
B2CF: 88         DEY
B2D0: F0 F8      BEQ $B2CA
B2D2: 30 C8      BMI $B29C

```

```

B2D4: 60         RTS
B2D5: A0 02      LDY #$02
B2D7: B1 70      LDA ($70),Y
B2D9: 85 74      STA $74
B2DB: C8         INY
B2DC: B1 70      LDA ($70),Y
B2DE: 85 75      STA $75
B2E0: A0 01      LDY #$01
B2E2: B1 72      LDA ($72),Y
B2E4: F0 26      BEQ $B30C
B2E6: C8         INY
B2E7: D8         CLD
B2E8: 18         CLC
B2E9: B1 72      LDA ($72),Y
B2EB: 65 74      ADC $74
B2ED: 91 72      STA ($72),Y
B2EF: C8         INY
B2F0: B1 72      LDA ($72),Y
B2F2: 65 75      ADC $75
B2F4: 91 72      STA ($72),Y
B2F6: B0 15      BCS $B30D
B2F8: EA         NOP
B2F9: EA         NOP
B2FA: EA         NOP
B2FB: EA         NOP
B2FC: A0 01      LDY #$01
B2FE: B1 72      LDA ($72),Y
B300: 48         PHA
B301: 88         DEY
B302: F0 FA      BEQ $B2FE
B304: 68         PLA
B305: 85 72      STA $72
B307: 68         PLA
B308: 85 73      STA $73
B30A: 90 D4      BCC $B2E0
B30C: 60         RTS
B30D: A9 0A      LDA #$0A
B30F: 85 74      STA $74
B311: 85 76      STA $76
B313: A9 00      LDA #$00
B315: 85 75      STA $75
B317: 85 77      STA $77
B319: A9 01      LDA #$01
B31B: 85 72      STA $72
B31D: A9 05      LDA #$05
B31F: 85 73      STA $73
B321: A0 01      LDY #$01
B323: B1 72      LDA ($72),Y
B325: F0 2B      BEQ $B352
B327: C8         INY
B328: A5 74      LDA $74
B32A: 91 72      STA ($72),Y
B32C: C8         INY
B32D: A5 75      LDA $75
B32F: 91 72      STA ($72),Y

```

```

B331: A0 01      LDY  #$01
B333: B1 72      LDA  ($72),
B335: 48         PHA
B336: 88         DEY
B337: F0 FA      BEQ  $B333
B339: 68         PLA
B33A: 85 72      STA  $72
B33C: 68         PLA
B33D: 85 73      STA  $73
B33F: 18         CLC
B340: A5 74      LDA  $74
B342: 65 76      ADC  $76
B344: 85 74      STA  $74
B346: A5 75      LDA  $75
B348: 65 77      ADC  $77
B34A: 85 75      STA  $75
B34C: B0 06      BCS  $B354
B34E: 90 D1      BCC  $B321
B350: EA        NOP
B351: EA        NOP
B352: EA        NOP
B353: EA        NOP
B354: 60        RTS
B355: 18        CLC
B356: A0 00      LDY  #$00

```

Print message

#7E: LSB Message location

#7F: MSB

```

B358: B1 7E      LDA  ($7E),Y
B35A: C9 FF      CMP  #$FF
B35C: F0 06      BEQ  $B364
B35E: 99 80 BB   STA  $BB80,Y
B361: C8         INY
B362: D0 F4      BNE  $B358
B364: 60        RTS
B365: EA        NOP
B366: EA        NOP
B367: EA        NOP
B368: EA        NOP
B369: EA        NOP
B36A: 41 50 50 45 4E 44
B370: 2F 52 45 4E 55 4D 20 60
B378: 44 2E 59 41 4E 20 31 39
B380: 38 34 FF 3F 53 49 5A 45
B388: 20 45 52 52 4F 52 20 20
B390: 20 20 20 20 20 20 20 20
B398: 20 20 20 FF 3F 41 50 50
B3A0: 45 4E 44 20 53 49 5A 45
B3A8: 20 45 52 52 4F 52 20 20
B3B0: 20 20 20 20 FF 3F 52 45
B3B8: 4E 55 4D 42 45 52 20 45
B3C0: 52 52 4F 52 20 20 20 20
B3C8: 20 20 20 20 20 FF 4F 4B
B3D0: 20 2D 49 4E 20 4D 45 4D

```

```

B3D8: 4F 52 59 20 20 20 20 20
B3E0: 20 20 20 20 20 20 FF
B3E7: EA        NOP
B3E8: EA        NOP
B3E9: EA        NOP
B3EA: A9 6A      LDA  #$6A
B3EC: 85 7E      STA  $7E

```

Message index/links to Print Messages

```

B3EE: A9 B3      LDA  #$B3
B3F0: 85 7F      STA  $7F
B3F2: 20 55 B3   JSR  $B355
B3F5: 60        RTS
B3F6: A9 83      LDA  #$83
B3F8: 85 7E      STA  $7E
B3FA: 20 55 B3   JSR  $B355
B3FD: 60        RTS
B3FE: A9 9C      LDA  #$9C
B400: 85 7E      STA  $7E
B402: 20 55 B3   JSR  $B355
B405: 60        RTS
B406: A9 B5      LDA  #$B5
B408: 85 7E      STA  $7E
B40A: 20 55 B3   JSR  $B355
B40D: 60        RTS
B40E: A9 CE      LDA  #$CE
B410: 85 7E      STA  $7E
B412: 20 55 B3   JSR  $B355
B415: 60        RTS
B416: EA        NOP

```

Program driver

```

B417: EA        NOP
B418: EA        NOP
B419: EA        NOP
B41A: EA        NOP
B41B: EA        NOP
B41C: EA        NOP
B41D: EA        NOP
B41E: EA        NOP
B41F: EA        NOP
B420: 20 03 B2   JSR  $B203
B423: 20 2A B2   JSR  $B22A
B426: B0 CE      BCS  $B3F6
B428: 20 50 B2   JSR  $B250
B42B: 90 E1      BCC  $B40E
B42D: 4C EA B3   JMP  $B3EA
B430: 20 03 B2   JSR  $B203
B433: 20 70 B2   JSR  $B270
B436: B0 C6      BCS  $B3FE
B438: 20 50 B2   JSR  $B250
B43B: 20 90 B2   JSR  $B290
B43E: 20 D5 B2   JSR  $B2D5
B441: 4C EA B3   JMP  $B3EA

```

Hints & Tips

BLOCK LINE DELETE

Getting rid of a chunk of unwanted program can be tedious if you simply type in line numbers and RETURN – particularly if lines are numbered irregularly. Here's a simple eight line program to take this drudgery off your hands.

It is entered in what will hopefully be a free portion of memory, starting at line 63325.

```
63325 X=1281 : REM START OF BASIC CHAIN
63326 IF DEEK(X+2)<>sss THEN X=DEEK(X): GOTO 63326
63327 Y = DEEK(X)
63328 IF DEEK(Y+2)<>fff THEN PRINT DEEK(Y+2);
        Y = DEEK(Y): GOTO 63328
63329 PRINT: INPUT "CONFIRM DELETION (Y/N)": A$
63330 IF A$ = "Y" THEN W = DEEK(X): DOKE X,Y:
        GOTO 63332
63331 END
63332 FOR Z = W TO Y-2: POKE Z,85: NEXT
63333 LIST
```

Atmos owners, with a debugged IF... THEN... ELSE can enter

```
63330 IF A$ = "Y" then W =
DEEK(X): DOKE X,Y: ELSE
END and delete 63331.
```

sss and fff are the markers for the block to be deleted. When the program is run, all line numbers higher than sss and lower than fff will be deleted. You'll have to enter the relevant line numbers when typing in the program. Alternatively you could insert lines asking for an input, together with checks to verify entered lines actually exist.

With the program entered, type RUN 63325. You'll be asked if you want confirmation or not, and the block deleted.

If you add any new lines after the block deletion, the deleted block will reappear as 28145 UUUUUUU... To clear this, enter 28145 and RETURN.

The Block Line Delete program works very simply. Here's what's happening.

63325 Sets X to start of Basic chain

63326 Checks line numbers for sss

63327 Checks line numbers for block to be deleted

63328 Checks line numbers for fff, where deletion stops

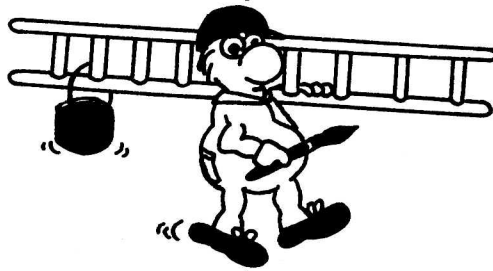
63329 Checks the range is correct

63330 Deletes the specified lines by updating BASIC pointers

63332 Fills deleted block with Us, thus destroying deleted lines and their pointers, so they aren't rediscovered if new lines are added

63333 Lists the final program
This little piece of Oric wizardry comes from C. R. Burnham.

VARIABLES SAVED



David Peat is just 14, but he's sent us three tips that really make us think he's an Oric whizz kid.

The first comes from his interest in l-o-n-g adventure programs. Even the Oric's memory isn't big enough to hold them all! So, he's had to store them on different tapes, and keeping variables and flags stored when loading different sections of the program is a problem. What David does is to POKE all the variables into an unused part of memory, load the new program and then PEEK them out again. Here's the POKE program, for eleven variables.

```
10 FOR N = 0 TO 10
20 POKE 1024 + N, A(N)
30 NEXT N
40 CLOAD(B$)
```

Variables are poked to locations from 1024. Increase the number of loops and you'll be able to store more variables.

B\$ is the name of the next part of the program. You must use brackets around the variable name. Now, start the second program section with the PEEK program.

```
10 FOR N = 0 TO 10
20 LET A(N) =
    PEEK(1024+N)
30 NEXT N
40 END
```

If you've saved the new program using AUTO, this first section will automatically run and the variables be read in automatically.

Tip number two relates to key pressed recognition numbers. Enter this two liner.

```
10 PRINT DEEK(783)
20 GOTO 10
```

RUN it and you'll get a listing of five figure numbers, the number depending on the key pressed. For example:

```
45310 No key pressed
48255 Cursor right
48319 Cursor down
48351 Cursor left
48375 Cursor up
48375 Space bar
and so on...
```

So far, interesting only.

However DEEK(783) can really speed up a game, like this...

POKE 755,X changes the speed at which the Oric scans the keys. X = 10 gives a rapid auto repeat and a slower program. X = 255 gives a slow auto repeat and a very fast program. However, POKE 755,X works only as long as no keyboard input is needed. Use DEEK(783) instead of KEY\$ and you can get a much faster games program. Use this method and you will more than halve the key response time in games. Who needs machine code?

David's third tip is an alternative to the CHR\$(27) method of changing text colour. Instead of CHR\$(27)"letter... characters" enter the following, and get the following effects.

```
PRINT CHR$(128) – black text
PRINT CHR$(129) – red text
PRINT CHR$(131) – yellow
    text
PRINT CHR$(132) – blue text
PRINT CHR$(133) – magenta
    text
PRINT CHR$(134) – cyan text
PRINT CHR$(135) – white
    text.
```

Numbers from 144 to 151 give background colours. This tip may not save time, but it certainly makes programs tidier through getting rid of attribute letters at the start of text or strings.

ERROR PEEK

A quick way to see all the Oric command words and error messages is to run this short program from fifteen year old M. J. Allen of Birmingham.

```
10 PTR = #C0EA
20 LAST = #C3CB
30 REPEAT
40 PRINT CHR$(PEEK(PTR));
50 IF PEEK(PTR) > 128 THEN
  PRINT
60 PTR = PTR + 1
70 UNTIL PTR = LAST
```

He found an unlisted command, EDIT. This acts as LIST but does not print Ready below the EDITed line and the cursor comes to rest next to the EDITed line.

When it comes to speeding up entry of programs, he has another little tip for entering random numbers. Entering a line such as

```
10 A = INT(RND(1)*9)+1
```

is tricky. A shorter way to do the same thing is this...

```
10 A% = RND(1)*10
```

The variable A% can be handled faster by the Oric, but should always be addressed as A%, not A.

Our Birmingham brain has also found a way to speed up

some sections of a BASIC program. Using the command

```
CALL #E6CA
```

turns off the interrupts and will make program sections run about 20% faster.

There are restrictions to this. Turning off the interrupts kills the keyboard, and the only way to revive it is to reset the computer. To restore the interrupts, use the command

```
CALL #E804
```

You'll need to use this before the program ends, or before the following commands: GET, INPUT, KEY\$, STOP and END - i.e. commands needing a keyboard output.

You'll also run into problems if you try to switch from TEXT to HIRES or vice versa with the interrupts off. Don't do this.

CALL #E804 affects both TEXT and HIRES screens. Locations to the left of the CAPS sign are effected in the text mode.

So, with careful use, CALL #E6CA could be a valuable aid to sorts, number crunching and so on.



COLOUR CODES

Achieving double height characters is through using CTRL D or CHR\$(4). You can change the colour of the characters of background or foreground with PAPER and INK commands. It can get confusing. Here's another way to do that, contributed by Mr T. Arnold of Abingdon, Oxfordshire.

It relies on one basic statement.

```
POKE #26A,67 : DOKE #26B,
#X : CLS
```

The value of X determines the effect. Here are some X values, and their effect.

- 010A Red, double height characters on black screen, until effect cancelled
- 020A Green characters, as above
- 030A Yellow characters, as above
- 040A Blue characters, as above
- 050A Magenta characters, as above
- 060A Cyan characters, as above
- 070A White characters, as above
- 110A White, double height characters on red screen, until cancelled
- 120A As above, green screen
- 130A As above, yellow screen

- 140A As above, blue screen
- 150A As above, magenta screen
- 160A As above, cyan screen
- 170A As above, white screen (you won't see a thing!)
- 180A Corrupts screen display, don't use

Other values give different results, so experiment for yourself.

The last X value brings us to Mr Arnold's second tip. If you either corrupt the display or crash and lock up the keyboard, try typing CALL 555, and RETURN. Usually the screen will be restored to PAPER 7, INK 0 and cleared, caps and keyclick on. It also clears all peeks, pokes and redefined characters. In other words, it acts as the fiddly reset button. So, remember, in an emergency, CALL 555.

This short program demonstrates the above tips and will let you try them out yourself.

```
10 INPUT X : REM ...
20 POKE #26A,67 : DOKE #
  26B,X : CLS
30 PRINT : PRINT : "ORIC
  TEST"
40 PRINT : PRINT : PRINT
50 IF X = 999 THEN CALL
  555
60 GOTO 10
```

KEYBOARD RESETTNG

Anyone who's cursed at the location of the reset button will be grateful for this tip from Mr Russell of Keighley in West Yorkshire. In one line, you can redefine the ! key to act as a reset.

Type in this...

```
DOKE #2F5, #F430
(RETURN)
```

Now, whenever you need to reset your Oric, press !, RETURN and you'll have an instantaneous reset. Now why couldn't the designers have done something simple like that?

52 Oric Owner





Totally devoted to you!

Oric Owner is the official magazine devoted to the Oric 1 and Atmos home computers.

It's crammed full of in-depth information, advance news on the latest add-ons, superb programs and interviews with the engineers who designed it.

The first issue is absolutely free when you buy your Oric, so why not keep ahead of the latest developments and subscribe to further issues. A years subscription of 6 issues is now only £7.50 (overseas p.o.a) so post the coupon today.

Can you imagine life without it?



ORIC OWNER SUBSCRIPTION FORM

Please send me the next 6 issues

I enclose a cheque for (cheques payable to Tansoft Ltd)

Name.....

Address.....

..... Postcode.....

Back issues are available for £1.20 each. If you missed your first free issue contact your dealer or Oric Products International who will supply you with one.

TANSOFT Tansoft Ltd, Units 1 & 2
Cambridge Techno Park Newmarket Road, Cambridge CB5 8PB

As market leaders in the ORIC Software field, IJK Software brought you these best sellers...



XENON-1

A true legend in computer software. This game topped the ORIC software charts in most high street stores during 1983, and became the overall best selling computer game in France for the same period, gaining rave reviews throughout the press.

"Uses the ORICS tremendous capabilities to excellent advantage...truly mind boggling...the graphics are absolutely superb, very smooth indeed...I can thoroughly recommend this program".

-WHAT MICRO, Sept. 1983

"This is the best arcade game yet available for the ORIC. The graphics and animation are superb".

-ORIC OWNER, Sept 1983

As fleet commander in the Xenon space academy, you must protect your home planet Radon from the Aards, Paratrons and the Zorgon battle star! This 100% machine code game features the smoothest, meanest arcade action around!

For the 48K ORIC-1 and ORIC ATMOS £8.50 inc.

ZORGONS REVENGE

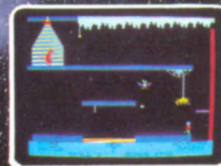
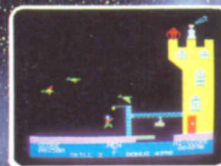
The second in the Xenon trilogy, written by the same author as Xenon-1, and continuing the high standard of excellence. Zorgons Revenge is well on its way to repeating the success of Xenon-1, and is again receiving rave reviews.

"This game certainly is the best arcade style effort I have yet seen on the ORIC. The graphics are superb...the author deserves praise...and I was impressed with the ease with which the game loaded, even though the cassettes contained 'fast' versions only". - PERSONAL COMPUTER WORLD GAMES SPECIAL, 1984

"The animation is smooth and detailed...this will be a solid gold hit". - COMPUTER CHOICE, Feb. 1984

"Some of the best graphics available for the ORIC...recommended". - PCG HIT, PERSONAL COMPUTER GAMES, Feb. 1984

"Quite the best game I've seen for the ORIC...superb graphics". - DAILY EXPRESS 10th Dec. 1983

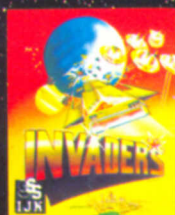


ALL OUR CURRENTLY AVAILABLE SOFTWARE WILL OPERATE ON BOTH THE ORIC-1 AND THE ORIC ATMOS

Following their defeat at the hands of the Xenon fleet the Zorgons have captured the Xenon princess Roz, and have imprisoned her in their castle. You are commissioned to rescue her by scouring the four corners of the Zorgon Empire to capture the magic stones. These stones, guarded by the Quadnogs, Terrapods and many other strange beasts, are needed to bridge the bottomless chasm surrounding the castle, enabling you to achieve your goal. Each one of the many varied stages in this scintillating 100% machine code mission will test your arcade ability as never before.

For the 48K ORIC-1 and ORIC ATMOS £8.50 inc.

OTHER TITLES IN THE IJK SOFTWARE ORIC RANGE...



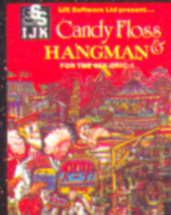
INVADERS
Machine code arcade game for 16K & 48K ORIC-1 and ORIC ATMOS
£7.50 inc.



FANTASY QUEST
Intriguing adventure for 48K ORIC-1 and ORIC ATMOS
£6.50 inc.



REVERSE
Superb version of the board game for 48K ORIC-1 and ORIC ATMOS
£6.50 inc.



CANDYFLOSS & HANGMAN
Two top educational programs for 48K ORIC-1 and ORIC ATMOS
£7.50 inc.



3D MAZE & BREAKOUT
Arcade action for 48K ORIC-1 and ORIC ATMOS
£7.50 inc.